UK Public Transport Information Profile

A TransXChange profile

for exchanging UK schedule data within the Bus Open Data Digital Service and similar systems

Document Control Log

Date	Modified By	Comment	Version
2020-04-13	SJR	Initial Draft	v0.1
2020-04-14	SJR	Added introduction, adjusted layout & pagination.	v0.2
2020-04-30	SR	Issued for consultation as v1.0	v1.0
2020-08-09	SJR	Amendments, expansions and clarifications following DfT review, including additional tables and figures and hence some renumbering.	v1.1
2021-06	TMR	Clarification and corrections following industry review and inclusion of BODS validator rules. Issued for further industry review.	V1.1a.4
2021-07-05	TMR	Final release following review	V1.1.A

Contents

List of Figures

List of Tables

1 Introduction

1.1 Scope

The Bus Open Data programme, coming from the Bus Services Act 2017, places a requirement on operators to produce TransXChange (TXC) files containing their timetables as open data. While TXC has been around for many years now, and is a mature standard, there are a large variety of ways in which data can be constructed within the standard. Some of these are more concise than others, and some inadvertently hide simple data in complex constructs. This makes it hard to interpret data, and users have to be able to interpret the full standard in order to accommodate the different ways that data may be written.

Furthermore, there is a high degree of inheritance of elements permissible within TXC, and this can lead to differences in interpretation, as different systems differ in which elements within a block remain as the default, and which are overridden. This can lead to errors which, when coupled with the complexity of some implementations, are then hard to identify and correct.

Additionally, the continued reliance by the industry on TXC v2.1 has led to a number of errors which are more avoidable if later versions are used. In particular, the coding of trips which run after midnight (operational day vs calendar day principles) is often wrong.

As part of the Bus Open Data Service (BODS) programme, the Department for Transport (DfT) has sponsored the creation of a TXC profile to support BODS. The aim of this profile is to specify a consistent use of elements and a consistent way of writing TXC that will be used within BODS and which will lead to a higher quality data set and, at the same time, lower the barriers to entry by users new to TXC.

This profile is set out in this document. Because it has a wider applicability than simply the BODS programme, the profile will be referred to as the public transport information profile. It will be abbreviated as the TXC-PTI profile, or simply TXC-PTI within this document.

1.2 Relationship to TransXChange

The TXC-PTI profile will be based on TransXChange version 2.4, as this is the updated version that has most traction and that will be specified in the BODS guidance. In particular, the TXC-PTI profile is based on the “general” schema and not the “registration” schema. This use of the general schema allows freedom to remove elements that would be mandatory in the registration schema, but which are not relevant to TXC-PTI.

It is important to note that the TXC-PTI profile is not a new version of the TXC standard. There is no new “XSD” schema definition file, and anyone creating a TXC v2.4 file will be able to make full use of the standard. All the elements within the schema that are mandatory in the schema will be mandatory in the TXC-PTI profile.

What the profile does, instead, is to specify how that standard is to be used. It specifies additional mandatory elements that are to be used, it specifies which data elements shall *not* be used, and it specifies which are optional. It also details in many instances how the different elements shall be populated, particularly in relation to identifying and versioning different data elements so as to be consistent.

Validation of data against the TXC-PTI profile will take place in two stages:

1) Is the data schema-compliant? This is a straightforward check that checks that the data as submitted is compliant with the basic requirements of the TXC v2.4 general schema. This is an in-built function of XML.
2) Is the data TXC-PTI compliant? This is a programmatic check to ensure that the data meets the requirements of this profile, carried out by analysing the data in the TXC

document. Bespoke systems are required to carry out these checks, and a validator has been developed by DfT as part of the BODS programme.

In addition to the validator checks BODS carries out data quality tests include some PTI-TXC profile related matters, but most are focused on improving the quality and accuracy of information rather than technical compliance. The Data Quality tests are described in the document found at:

https://publish.bus-data.dft.gov.uk/guidance/data-quality-definitions/

The PTI-TXC profile is an additional set of constraints and clarifications that sit on top of TXC version 2.4. While every care has been taken to ensure compliance with TXC, this document will necessarily need to be read alongside the TransXChange schema guidance document. The most recent published version of this is version 59, which contains information on all versions up to and including TXC v2.5. Elements introduced for v2.5 can be ignored, but users should note that the guidance also contains clarifications and corrections to earlier versions of the schema guidance / standard.

1.3 Abbreviations

BODS Bus Open Data Service. The DfT Bus Open Data Service programme emerging from Section 18 of the Bus Services Act 2017.

DfT The UK Department for Transport.

EBSR Electronic Bus Service Registration

OTC Office for the Traffic Commissioner. Responsible for the registering and compliance monitoring of local bus services to ensure that operators meet their statutory obligations.

NOC National Operator Code. A unique code, usually four characters, that identifies an operator or operating division of an operator and which should be consistent across all systems. This can be found in the NOC database owned and managed by Traveline.

TIL Traveline Information Limited. A company with overall responsibility for traveline-branded public transport information services in the UK, widely advertised on bus stops and buses.

TXC TransXChange. The UK *de facto* standard for timetable interchange.

TXC-PTI The TXC Public Transport Information profile, set out in this document.

XML E**X**tensible **M**arkup **L**anguage. The technology used to encode TXC documents.

1.4 Structure of this document

This document is split into sections which form a logical walk through the TXC schema. The sections are not intended to replicate the order that elements occur within a TXC document.

Examples are provided wherever appropriate. As far as possible these are based on real-world examples but should not be taken as representing the operation of actual services or organisations.

Within the document, summaries are placed at key points. These summaries indicate which are the key mandatory items and which are the optional items. They also highlight any key restrictions identified in the text. Summaries are indicated using coloured icons, and these have the following meanings:

➡	This summarises key mandatory points from the text.	➡	This summarises key optional points from the text

It must be emphasised that, with optional elements, there is no requirement to include them in a TXC-PTI document although they are strongly recommended in many instances. The specification nevertheless sets out how these optional elements should be populated, including mandatory sub-elements, when the optional element is present.

The use of ‘Not used’ when describing an element or enumeration signifies that the element should not be used when providing TXC-PTI data

The use of Optional means that the data can be provided, but that any implementation using the TXC-PTI profile will not require the element and consumers cannot rely on its availability.

The TXC-PTI profile is being used by BODS, this has several implementation specific requirements. In addition, the BODS implementation carries out validation of data supplied to the service, some of these validation tests and the results benefit from additional explanation. In both situations they are summarised in this document using a bus icon:

🚌	This summarises key implementation points for the BODS.

1.5 Credits

This TXC-PTI profile has been sponsored by the DfT, with extensive contributions from the devolved administrations in Scotland and Wales, the UK bus industry, local transport Authorities, and the UK public transport information community. Their input is gratefully acknowledged.

2 General

2.1 General principles

The TXC-PTI profile has been written with a specific set of principles in mind. The aim is to create a simplified profile, which can be used consistently by all systems / operators and where data is clear and unambiguous data. This will lead to data that is more readily understood by downstream open data users, will lead to a higher take up, easier diagnosis, and hence better applications.

The overriding principle is that data should be stated, once, where is best placed. In TXC there is too much scope for writing a mixture of elements, some of which are inherited, some of which are not. There is merit in re-using data where possible, because this keeps file sizes down, but far too often data has been presented where only some of the information in a particular block has been inherited while some has been overridden. This is particularly true of operating profiles, which define the days / dates when the trips actually run.

A secondary, although still important, principle is that data should be stated unambiguously. There are many elements where, for largely historic reasons, ambiguity or shades of meaning have crept in. TXC-PTI aims to remove this ambiguity so that it is clear what the intent of a set of elements or a particular block intend. Again, this is something which is particularly true of operating profiles although it is by no means the only one.

Finally, a third general principle is that ids shall be provided and defined which a) are persistent from one supply of a TXC-PTI file for given service/line to the next; b) provide uniqueness to any given service and line, such that line 8 in one locality is different from line 8 operated by the same operator in a different locality; and c) make it clear when a particular line is changing from one version to the next, and which line it is that is changing.

The implementation of these principles will be described more fully in the sections that follow.

2.2 Top Level Elements

The table below gives an overview of the usage of the top level elements (those immediately below the root **TransXChange** element) within the TXC general schema. For more details see the sections referenced.

Table 1 - Usage of top-level elements in TXC-PTI

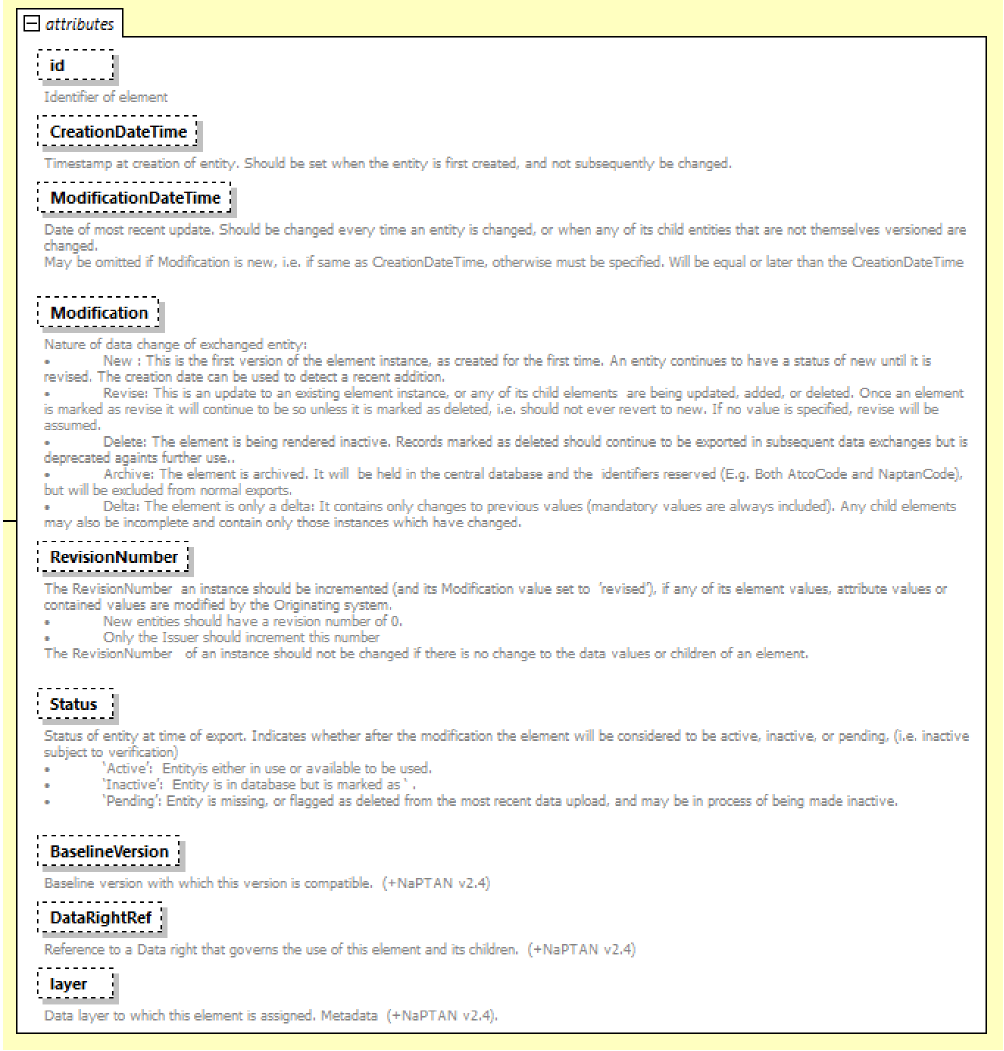
Element Name	**Data Type**	**Used in TXC-PTI**
ServicedOrganisations	*ServicedOrganisationsStructure*	Where appropriate. See section 3.
ServiceCalendars	*ServiceCalendarsStructure*	Not used in TXC-PTI
NptgLocalities	*TransXChangeNptgLocalitiesStructure*	Not used in TXC-PTI
StopPoints	*TransXChangeStopPointsStructure*	Yes. Shall be present. See section 6.
StopAreas	*StopAreasStructure*	Not used
RouteSections	*RouteSectionsStructure*	Yes. Shall be present. See section 7.3.
Routes	*RoutesStructure*	Yes. Shall be present. See section 7.2.
JourneyPatternSections	*JourneyPatternSectionsStructure*	Yes. Shall be present. See section 8.3.
Operators	*OperatorsStructure*	Yes. Shall be present. See section 4.

Services	*ServicesStructure*	Yes. Shall be present. See section 5.
VehicleJourneys	*VehicleJourneysStructure*	Yes. Shall be present. See section 9.
Registrations	*RegistrationsStructure*	Not used in TXC-PTI
SupportingDocuments	*SupportingDocumentsStructure*	Not used in TXC-PTI

2.3 Versions

A common problem with TXC documents up to the present has been the incorrect use of codes to encode versioning information within a file, despite TXC elements generally being provided with a set of attributes which allow this information to be correctly represented. This can lead to, for example, confusion as to whether a file contains a new service or simply a new version of an existing service.

TXC-PTI has adopted some general rules about versioning which shall be followed in those elements which support it. The general principle shall be that, wherever possible, ids and codes used within a series of documents to describe a service shall be consistent from document to document, and that sequences of documents shall be identified and tracked through the documents' version control attributes.



attributes

id

Identifier of element

CreationDateTime

Timestamp at creation of entity. Should be set when the entity is first created, and not subsequently be changed.

ModificationDateTime

Date of most recent update. Should be changed every time an entity is changed, or when any of its child entities that are not themselves versioned are changed.
May be omitted if Modification is new, i.e. if same as CreationDateTime, otherwise must be specified. Will be equal or later than the CreationDateTime

Modification

Nature of data change of exchanged entity:

- New : This is the first version of the element instance, as created for the first time. An entity continues to have a status of new until it is revised. The creation date can be used to detect a recent addition.
- Revise: This is an update to an existing element instance, or any of its child elements are being updated, added, or deleted. Once an element is marked as revise it will continue to be so unless it is marked as deleted, i.e. should not ever revert to new. If no value is specified, revise will be assumed.
- Delete: The element is being rendered inactive. Records marked as deleted should continue to be exported in subsequent data exchanges but is deprecated againts further use..
- Archive: The element is archived. It will be held in the central database and the identifiers reserved (E.g. Both AtcoCode and NaptanCode), but will be excluded from normal exports.
- Delta: The element is only a delta: It contains only changes to previous values (mandatory values are always included). Any child elements may also be incomplete and contain only those instances which have changed.

RevisionNumber

The RevisionNumber an instance should be incremented (and its Modification value set to 'revised'), if any of its element values, attribute values or contained values are modified by the Originating system.

- New entities should have a revision number of 0.
- Only the Issuer should increment this number

The RevisionNumber of an instance should not be changed if there is no change to the data values or children of an element.

Status

Status of entity at time of export. Indicates whether after the modification the element will be considered to be active, inactive, or pending, (i.e. inactive subject to verification)

- 'Active': Entityis either in use or available to be used.
- 'Inactive': Entity is in database but is marked as ' .
- 'Pending': Entity is missing, or flagged as deleted from the most recent data upload, and may be in process of being made inactive.

BaselineVersion

Baseline version with which this version is compatible. (+NaPTAN v2.4)

DataRightRef

Reference to a Data right that governs the use of this element and its children. (+NaPTAN v2.4)

layer

Data layer to which this element is assigned. Metadata (+NaPTAN v2.4).

Figure 1 - An indicative set of element attributes

- *CreationDateTime* shall be populated with the date and time at which the data was first created and shall not be changed in any future version of the same element.
- *ModificationDateTime* shall be populated with the date and time of the current data. It shall be omitted for the original (version 0) issue of the data but shall be present in all subsequent versions of the same data.
- *Modification* is a status flag and shall be set to either “New” for the first issue of the data, or to “Revise” for subsequent versions of the same data. No other *Modification* enumerators shall be used.
- *RevisionNumber* is a sequence number that shall be 0 for the first issue of the data and that shall increase for each subsequent version of the data. It is important to note that this revision number is used to track TXC-PTI data from version to version and is independent of any other versioning system. For example, an operator may wish to update operational dates to accommodate school term changes. This would require a change of revision number in the TXC-PTI data, but would be independent of, and not reflect the current value of, the revision number recognised by OTC in respect of the service registration.
- *Status* is of limited value since data elements (e.g. **Service/OperatingProfile**) will describe the extent to which data is active. There is therefore no requirement to provide this attribute, and for any conflicts between *Status* and data, the *Status* flag shall be ignored.

These are summarised in the table below:

Table 2 - Attributes used for versioning

Attribute Name	**Data Type**	**Used in TXC-PTI**
CreationDateTime	*dateTime*	Yes. Always present and does not change. The date and time of revision number 0.
ModificationDateTime	*dateTime*	Yes, for all revision numbers other than 0. Omitted otherwise. The date and time of this revision.
Modification	*ModificationEnumeration*	Yes. Permitted values are: • New • Revise
RevisionNumber	*RevisionNumberType*	Yes. First version is revision number 0. Increment by 1 for each revision thereafter.
Status	*StatusEnumeration*	Optional

In all cases where it is supported by the element, version control shall be implemented in element attributes.

Version control where data for a Service is split into multiple TXC files can become complex to manage very quickly both for data provider and consumer. It is recommended that data for each version of a Service is provided in a single TXC file to reduce complexity.

The validator tests for:

- CreationDateTime being always present and that it remains the same between files submitted with different RevisionNumber.
- ModificationDateTime being newer than CreationDateTime if RevisionNumber is greater than 0.
- Modification is a valid value.
- RevisionNumber for new revisions is larger than the existing published one.

When a new file is supplied to BODS with a new RevisionNumber the previous RevisionNumber file is superseded from the publishing date of the new file on BODS.

New higher RevisionNumber files supersede lower RevisionNumber files in their entirety,

2.3.1 Versioning in BODS

Examples of the use of RevisionNumber for versioning in BODS are provided below.

The initial file provided to BODS for a service would have a RevisionNumber = 0, in this example the service starts on 1 February 2021 and has no end date.

Revision Number	Start Date	End Date	February 2021	March 2021	April 2021	May 2021
0	01-Feb-21	(empty)	🟦	🟦	🟦	🟦

When there is a change to the service a new file is provided to BODS with a RevisionNumber = 1. This change starts on 1 March 2021.

Revision Number	Start Date	End Date	February 2021	March 2021	April 2021	May 2021
1	01-Mar-21	(empty)		🟧	🟧	🟧

This file because it has a RevisionNumber = 1 which is greater than the previous RevisionNumber = 0 will become the published data on BODS for the service. The original file with RevisionNumber = 0 will no longer be published. If this file was published on BODS on 14 February 2021, then from 14 February to 28 February there would be no service data live.

To update data for a Service in advance of a change and to retain the previous operating information two files need to be published to BODS, provided in a ZIP file. This file is published on BODS on 14 February 2021.

Revision Number	Start Date	End Date	February 2021	March 2021	April 2021	May 2021
0	01-Feb-21	(empty)	🟦	🟦	🟦	🟦
1	01-Mar-21	31-Mar-21		🟧	🟧	🟧

Both files RevisionNumber = 0 and RevisionNumber = 1 will be published in BODS. As a new revision starting on 1 March 2021 is also published and has a new

ModificationDatetime, it needs to have a revision number greater than the previous RevisionNumber = 0.

The file with RevisionNumber = 1 has a start date of 1 March 2021.

Data consumers will import the current published files from BODS and will need to import both files to identify that whilst the file with RevisionNumber = 1 has the greatest RevisionNumber the StartDate in the file is not valid until 1 March 2021. The data consumer then checks the file with RevisionNumber = 0 and identifies that the data is valid. On the 1 March 2021 the RevisionNumber = 1 becomes valid and supersedes the file RevisionNumber = 0.

The RevisionNumber = 0 file will continue to be published along with RevisionNumber = 1 until a new ZIP file with a new RevisionNumber is published.

In the situation where there is a short term change to a service and the service reverts to the previous operation following that change data needs to be provided as follows. Both RevisionNumber = 0 and RevisionNumber = 1 must be provided in the same zip file to BODS.

Revision Number	Start Date	End Date	February 2021	March 2021	April 2021	May 2021
0	01-Feb-21	(empty)	🟦	🟦	🟦	🟦
1	01-Mar-21	31-Mar-21		🟧		

Both files RevisionNumber = 0 and RevisionNumber = 1 will be published in BODS.

The file with RevisionNumber = 1 has a start date of 1 March 2021.

Data consumers will import the current published files from BODS and will need to import both files to identify that whilst the file with RevisionNumber = 1 has the greatest RevisionNumber the StartDate in the file is not valid until 1 March 2021. The data consumer then checks the file with RevisionNumber = 0 and identifies that the data is valid. On the 1 March 2021 the RevisionNumber = 1 becomes valid and supersedes the file RevisionNumber = 0.

On the 1 April 2021 RevisionNumber = 1 contains no valid data, the data consumer would then check the file RevisionNumber = 0 and find that it contains valid data.

In the event that a further change to the service is being made two files, RevisionNumber = 0 and RevisionNumber = 2 need to be published to BODS, provided in a ZIP file. This file is published on BODS on 12 April 2021.

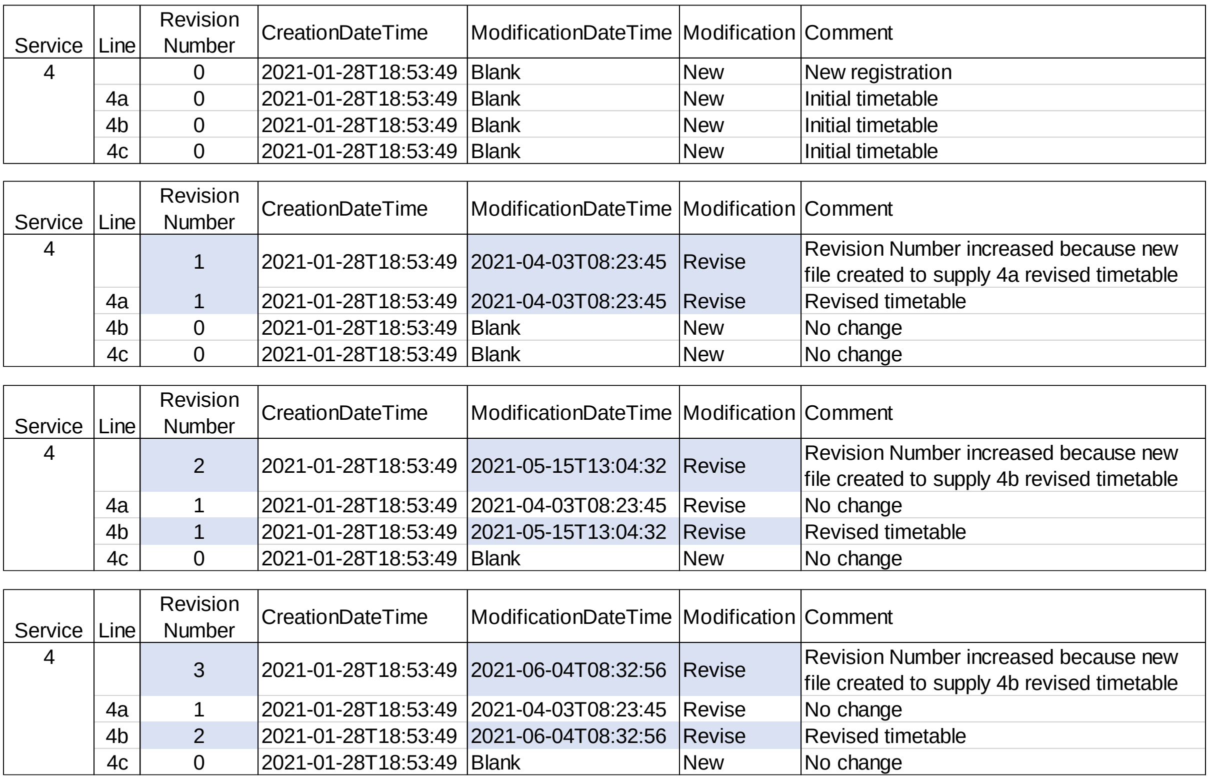
Revision Number	Start Date	End Date	February 2021	March 2021	April 2021	May 2021
0	01-Feb-21	(empty)	🟦	🟦	🟦	🟦
2	01-Jun-21	(empty)				🟩

Both files RevisionNumber = 0 and RevisionNumber = 2 will be published in BODS. The file with RevisionNumber = 1 will no longer be published.

The data consumer needs to identify that RevisionNumber = 2 becomes valid at its start date of 1 May 2021 and supersedes RevisionNumber = 0 data. The RevisionNumber = 0 file will continue to be published along with RevisionNumber = 2 until a new ZIP file with a new RevisionNumber is published.

In addition to version control within BODS TransXChange supports versioning through RevisionNumber at lower levels, for example within individual lines, where data for one or more Line is being updated. The use of RevisionNumber is helpful to data consumers to identify which parts of a Service have changed.

The following tables show how the revision numbers for a Service may increment if changes are made over time to each associated Line.



Service	Line	Revision Number	CreationDateTime	ModificationDateTime	Modification	Comment
4		0	2021-01-28T18:53:49	Blank	New	New registration
	4a	0	2021-01-28T18:53:49	Blank	New	Initial timetable
	4b	0	2021-01-28T18:53:49	Blank	New	Initial timetable
	4c	0	2021-01-28T18:53:49	Blank	New	Initial timetable

Service	Line	Revision Number	CreationDateTime	ModificationDateTime	Modification	Comment
4		1	2021-01-28T18:53:49	2021-04-03T08:23:45	Revise	Revision Number increased because new file created to supply 4a revised timetable
	4a	1	2021-01-28T18:53:49	2021-04-03T08:23:45	Revise	Revised timetable
	4b	0	2021-01-28T18:53:49	Blank	New	No change
	4c	0	2021-01-28T18:53:49	Blank	New	No change

Service	Line	Revision Number	CreationDateTime	ModificationDateTime	Modification	Comment
4		2	2021-01-28T18:53:49	2021-05-15T13:04:32	Revise	Revision Number increased because new file created to supply 4b revised timetable
	4a	1	2021-01-28T18:53:49	2021-04-03T08:23:45	Revise	No change
	4b	1	2021-01-28T18:53:49	2021-05-15T13:04:32	Revise	Revised timetable
	4c	0	2021-01-28T18:53:49	Blank	New	No change

Service	Line	Revision Number	CreationDateTime	ModificationDateTime	Modification	Comment
4		3	2021-01-28T18:53:49	2021-06-04T08:32:56	Revise	Revision Number increased because new file created to supply 4b revised timetable
	4a	1	2021-01-28T18:53:49	2021-04-03T08:23:45	Revise	No change
	4b	2	2021-01-28T18:53:49	2021-06-04T08:32:56	Revise	Revised timetable
	4c	0	2021-01-28T18:53:49	Blank	New	No change

2.4 Accessibility information

2.4.1 Introduction

Early versions of TransXChange schemas had no facility for adding information about vehicles, particularly with respect to wheelchair accessibility and on-board facilities for passengers with visual and/or other impairments. TXC version 2.4 was the first version to include this information, and this was then later extended by TXC version 2.5. Despite the full set of accessibility parameters only being available in v2.5, TXC-PTI can nevertheless make use of the facilities that are in v2.4 to describe the type of vehicle and some of the accessibility features of that vehicle.

Providing information about the accessibility features of the vehicle is optional in TXC-PTI, but is strongly recommended.

Irrespective of the data contained in a TXC-PTI document, there is no obligation on the operator, beyond those set out in law, to provide any of the equipment listed. Operational issues will determine whether any particular trip will operate as stated. The best that the TXC-PTI document can do is indicate what trips are likely to be available to people with reduced mobility or other impairments. As a consequence, the fact that an operator cannot guarantee whether or not a trip is wheelchair accessible (say) should not prevent them supplying data to say that it generally is.

2.4.2 The VehicleType element

Information about the accessibility features of a vehicle is achieved through the **VehicleType** element. This is constructed as shown below.

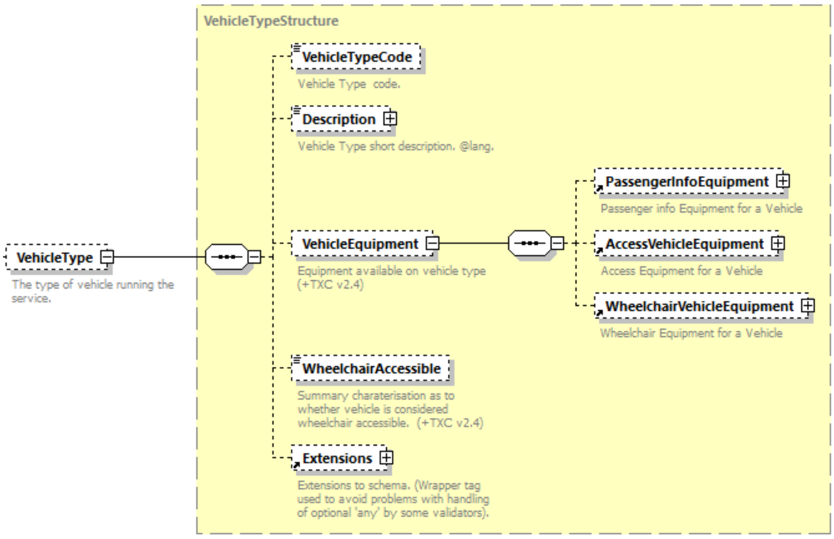

VehicleTypeStructure
VehicleTypeCode
Vehicle Type code.
Description
Vehicle Type short description. @lang.
PassengerInfoEquipment
Passenger info Equipment for a Vehicle
VehicleType
The type of vehicle running the service.
VehicleEquipment
Equipment available on vehicle type (+TXC v2.4)
AccessVehicleEquipment
Access Equipment for a Vehicle
WheelchairVehicleEquipment
Wheelchair Equipment for a Vehicle
WheelchairAccessible
Summary charaterisation as to whether vehicle is considered wheelchair accessible. (+TXC v2.4)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 2 - The VehicleType structure

Each vehicle may have a **VehicleTypeCode** and a **Description**; these are operator-defined elements that describe the vehicle. For example, “LFDD” might be the code, and “Low Floor Double Decker” the description. These are textual elements and are only used for visual reference, rather than any direct use in e.g. journey planning, and so can be omitted.

➡ **VehicleTypeCode** and **Description** are optional within **VehicleType**.

Each vehicle may also have a **VehicleEquipment** element. This element contains sub-elements which provide details about the equipment available for passenger information (**PassengerInfoEquipment**), the information regarding the accessibility of the vehicle itself, such as whether low floor or fitted with a ramp (**AccessVehicleEquipment**), and further information on wheelchair spaces, including how many and whether they need pre-booking and how (**WheelchairVehicleEquipment**). These three sub-elements are described more fully in the sections below.

➡ The main purpose of including a **VehicleType** element in TXC-PTI is to include information about the vehicle. Within TXC-PTI, therefore, where a **VehicleType** is provided it shall include a **VehicleEquipment** element, with at least one of the **PassengerInfoEquipment**, **AccessVehicleEquipment** or **WheelchairVehicleEquipment** sub-elements.

Finally, the **VehicleType** element also has a Boolean true/false flag, **WheelchairAccessible**, which states in simple terms whether the vehicle is accessible to passengers in a wheelchair. This flag should be stated explicitly in order to avoid doubt over the meaning of its omission. Note that, given the requirements of the Public Service Vehicle Accessibility Regulations 2000, as of 1st January 2020 it is expected that most vehicles will have this flag set to true.

➡ In TXC-PTI, the **VehicleType** element shall have a **WheelchairAccessible** flag which shall be set as appropriate. For most vehicles this will be true.

2.4.3 Passenger information equipment

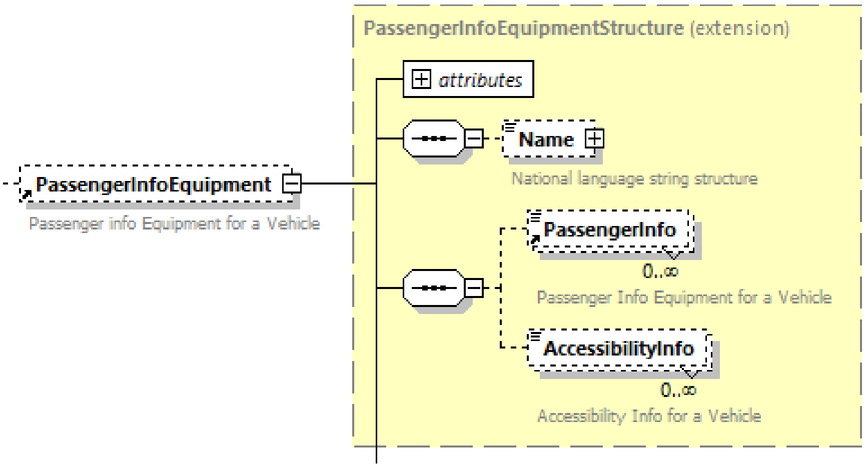

PassengerInfoEquipmentStructure (extension)
attributes
Name
National language string structure
PassengerInfoEquipment
Passenger info Equipment for a Vehicle
PassengerInfo
0..∞
Passenger Info Equipment for a Vehicle
AccessibilityInfo
0..∞
Accessibility Info for a Vehicle

Figure 3 - The PassengerInfoEquipment element

Passenger information equipment describes, through a set of elements which are classified as either passenger or accessibility information, the facilities that the vehicle has for presenting passenger information. The allowable values are shown in Table 3 below.

Table 3 - Permitted values for PassengerInfoEquipment

Group	Permitted Values	Used in TXC-PTI
PassengerInfo	nextStopIndicator	Where appropriate
	stopAnnouncements	Where appropriate
	passengerInfoFacility	Optional
	Other	Optional
AccessibilityInfo	audioInformation	Where appropriate
	audioForHearingImpaired	Where appropriate
	visualDisplays	Where appropriate
	displaysForVisuallyImpaired	Where appropriate
	tactilePlatformEdges	Optional
	tactileGuidingStrips	Optional
	largePrintTimetables	Optional
	other	Optional

For example, a vehicle which has a “next stop” indicator and audio information might be described as follows:

```
<PassengerInfoEquipment>
  <PassengerInfo>nextStopIndicator</PassengerInfo>
  <PassengerInfo>stopAnnouncements</PassengerInfo>
  <AccessibilityInfo>audioInformation</AccessibilityInfo>
  <AccessibilityInfo>audioForHearingImpaired</AccessibilityInfo>
</PassengerInfoEquipment>
```

2.4.4 Vehicle access equipment

The **AccessVehicleEquipment** element describes how the vehicle may be accessed. A number of elements, such as the gap to the platform are not especially useful in a bus context, while others such as low floor and ramp flags are important. Their usage for TXC-PTI is shown in Table 4 below.

Note that for both LowFloor and Ramp, it should be stated explicitly whether these features are present or not. This avoids any doubt which may arise from their omission.

Table 4 – Permitted elements for AccessVehicleEquipment

Element Name	Data Type	Used in TXC-PTI
LowFloor	*boolean*	Yes. State explicitly whether true or false.
Ramp	*boolean*	Yes. State explicitly whether true or false.
RampBearingCapacity	*Weight*	Optional

NumberOfSteps	*nonNegativeInteger*	Where appropriate
BoardingHeight	*Length*	Optional
GapToPlatform	*Length*	Optional
WidthOfAccessArea	*Length*	Optional
HeightOfAccessArea	*Length*	Optional
AutomaticDoors	*boolean*	Optional

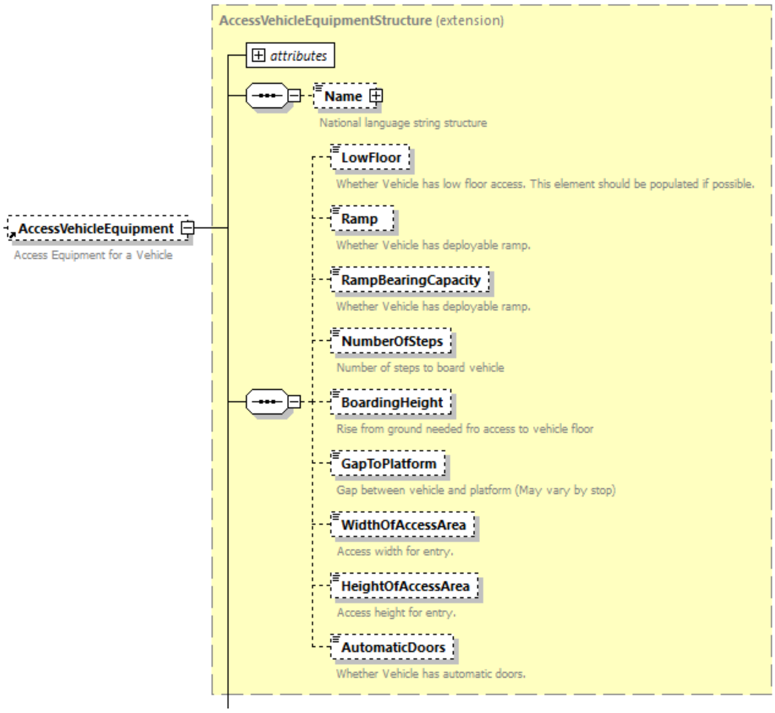

AccessVehicleEquipmentStructure (extension)
attributes
Name
National language string structure
LowFloor
Whether Vehicle has low floor access. This element should be populated if possible.
AccessVehicleEquipment
Access Equipment for a Vehicle
Ramp
Whether Vehicle has deployable ramp.
RampBearingCapacity
Whether Vehicle has deployable ramp.
NumberOfSteps
Number of steps to board vehicle
BoardingHeight
Rise from ground needed fro access to vehicle floor
GapToPlatform
Gap between vehicle and platform (May vary by stop)
WidthOfAccessArea
Access width for entry.
HeightOfAccessArea
Access height for entry.
AutomaticDoors
Whether Vehicle has automatic doors.

Figure 4 - The AccessVehicleEquipment element

For example, a vehicle which has a low floor, but which does not have a deployable ramp for wheelchair access, would be coded as:

```
<AccessVehicleEquipment>
   <LowFloor>true</LowFloor>
   <Ramp>false</Ramp>
</AccessVehicleEquipment>
```

2.4.5 Wheelchair-specific information

The WheelchairVehicleEquipment element provides information on the number and size of wheelchair spaces in the vehicle, together with information on whether or not advance booking is required, and the number to call if it is. Their usage for TXC-PTI is shown in Table 5 below.

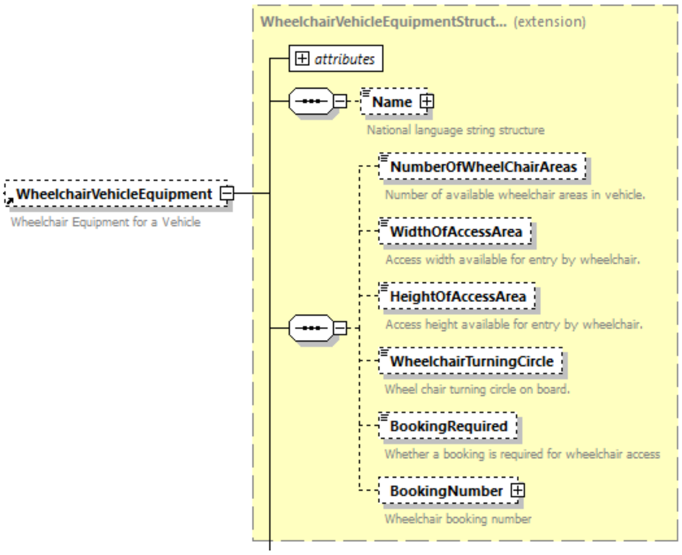

WheelchairVehicleEquipmentStruct... (extension)
attributes
Name
National language string structure
NumberOfWheelChairAreas
Number of available wheelchair areas in vehicle.
WidthOfAccessArea
Access width available for entry by wheelchair.
HeightOfAccessArea
Access height available for entry by wheelchair.
WheelchairTurningCircle
Wheel chair turning circle on board.
BookingRequired
Whether a booking is required for wheelchair access
BookingNumber
Wheelchair booking number
WheelchairVehicleEquipment
Wheelchair Equipment for a Vehicle

Figure 5 - The WheelchairVehicleEquipment element

Table 5 - Permitted elements for WheelchairVehicleEquipment

Element Name	Data Type	Used in TXC-PTI
NumberOfWheelChairAreas	*nonNegativeInteger*	Where appropriate
WidthOfAccessArea	*Length*	Optional
HeightOfAccessArea	*Length*	Optional
WheelchairTurningCircle	*Length*	Optional
BookingRequired	*boolean*	Where appropriate
BookingNumber	*TelephoneContactStructure*	Yes, if BookingRequired is true

For example, a vehicle with 2 wheelchair spaces which require booking may be coded as follows:

```
<WheelchairVehicleEquipment>
   <NumberOfWheelChairAreas>2</NumberOfWheelChairAreas>
   <BookingRequired>true</BookingRequired>
   <BookingNumber>
   <TelNationalNumber>01234 567890</TelNationalNumber>
   </BookingNumber>
</WheelchairVehicleEquipment>
```

2.5 Notes

Notes are permitted in various places in TXC-PTI and follow the same structure as in a standard TXC document.

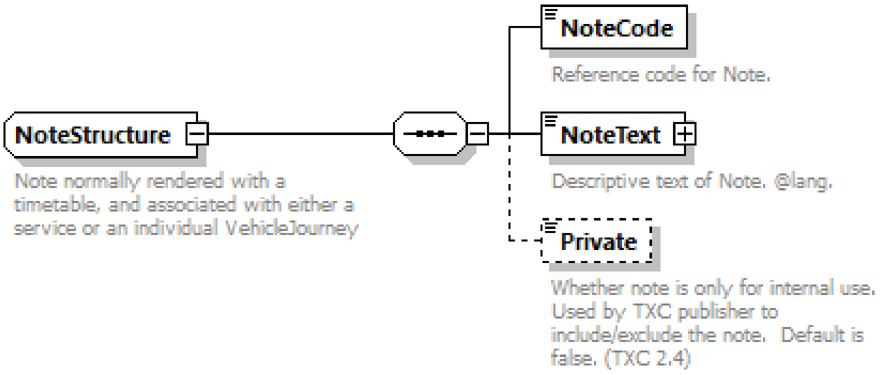

NoteCode
Reference code for Note.
NoteStructure
Note normally rendered with a timetable, and associated with either a service or an individual VehicleJourney
NoteText
Descriptive text of Note. @lang.
Private
Whether note is only for internal use. Used by TXC publisher to include/exclude the note. Default is false. (TXC 2.4)

Figure 6 - The Note structure

There are no specific requirements for notes that arise within TXC-PTI, except that:

a) notes should not be flagged as private; and

b) notes should not encapsulate anything in a text format that is best described through data. This is because consuming systems cannot process textual data without a high degree of effort.

An example of an inappropriate note might include, for example, “Starts 3 minutes earlier at Cefn Coch Inn”, as this should be encoded with suitable stop and timing records.

For TXC-PTI, **Notes** are optional.

For TXC-PTI, where **Notes** are used, then these shall not encapsulate information that is best described through data.

The validator tests for dates within the Notes fields and flags a warning if any found as date related data should be properly coded.

Data in Notes should not include any characters or constructs that could be used for SQL injections or other similar purposes. Please do not include any of the following characters in the field: , [ ] { } ^ = @ : ; # $ £ ? % + < > « » \ / | ~ _ ¬

If Notes are used, private element should only be false. If set to true the validator will flag this as an error.

3 Serviced organisations

3.1 Introduction

TXC has a mechanism for holding the dates when organisations, such as schools, are open and closed. This is the **ServicedOrganisation**. This gives one place where a common set of dates can be defined, and these can then be applied to individual vehicle journeys by a straightforward reference to the serviced organisation code. A **ServicedOrganisation** also contains a code by which the organisation can be referred, and a description.

This makes a serviced organisation very useful in terms of individual trips. Rather than coding them as operating date records in a **VehicleJourney**, reference to the **ServicedOrganisation** allows them to be given context when presenting them to users either as part of a journey plan or via a traditional timetable, for example "operates on Dame Alice Harpur school days only".

For this reason, TXC-PTI encourages the use of serviced organisations.

However, operators have expressed strong reservations about their ability to manage dates for each organisation as, while there may well be a core of commonality, many different schools and colleges will have different starts and ends to terms and will have different inset (non-teaching) days. Several schools and colleges may be featured on the same trip, and this additionally gives operators difficulties in knowing how to capture the description of when a particular trip runs; it may well be the case that, while one organisation is still open, another has already closed.

Nevertheless, an operator must know whether they are operating a particular trip or not, even if they cannot describe it through a **ServicedOrganisation**.

For TXC-PTI, then, a **ServicedOrganisation** is encouraged for the context that it can add, but is not mandatory. However, all trips shall, either through serviced organisations or through dated records, carry the correct dates so that it is unambiguous whether a trip runs or not on any particular day.

In TXC-PTI, the use of **ServicedOrganisations** is strongly recommended but is not mandatory. It is permitted to use a combination of **ServicedOrganisations** and dated records within the same TXC-PTI file.

In TXC-PTI, all trips shall be coded to state unambiguously when they operate. If **ServicedOrganisations** are not used, then appropriate date records shall be used.

It is important to provide accurate information that is kept up to date about the organisations served to ensure passengers know when a service is running. This requirement is backed by regulation, the Public Service Vehicles (Open Data) (England) Regulations 2020 require 'where the service is, or is to be, provided for the purpose of serving a school, college or other educational establishment, the dates of terms for that school, college or other educational establishment;'.

3.2 Defining a serviced organisation

A **ServicedOrganisation** has, at first glance, a significant number of elements that can be populated (see Figure 7). TXC-PTI will not use most of these, however, as they are of little use to the public; they carry such information as contact details and classification of a single school, or refer to NaPTAN or NPTG codes that are not public-facing.

Permitted elements and their values in TXC-PTI are as shown in the table below:

Table 6 – Permitted elements for ServicedOrganisation

Element Name			**Data Type**	**Used in TXC-PTI**
OrganisationCode			*OrganisationRefStructure*	Yes. Mandated by schema
PrivateCode			*normalizedString*	Where appropriate
Name			*NaturalLanguageString*	Yes. Meaningful name of organisation e.g. “College of West Anglia”, “Staffordshire schools”
PostalAddress			*UKPostalAddressStructure*	Optional
ContactTelephoneNumber			*TelephoneContactStructure*	Optional
ContactPerson			*normalizedString*	Optional
EmailAddress			*EmailAddressType*	Optional
WebSite			*anyURI*	Optional
FurtherDetails			*normalizedString*	Optional
ServicedOrganisationClassification			*ServicedOrganisationClassificationEnumeration*	Where appropriate
NatureOfOrganisation			*NaturalLanguageString*	Optional
PhaseOfEducation			*NaturalLanguageString*	Optional
WorkingDays			*DatePatternStructure*	Yes. Shall be included
	DateRange	StartDate	*StartDateType*	Yes. State explicit start date
		EndDate	*EndDateType*	Yes. State explicit end date. Shall not be omitted.
		Description	*NaturalLanguageString*	Where appropriate
		Provisional	*boolean*	Where appropriate. True for provisional dates, omitted otherwise.
		DateClassification	*DatePatternClassificationEnumeration*	Not used
Holidays			*DatePatternStructure*	Must not be used. All days which are not defined as working days are considered holidays.
ParentServicedOrganisationRef			*ServicedOrganisationRefStructure*	Optional
AdministrativeAreaRef			*AdministrativeAreaRefStructure*	Optional
NptgDistrictRef			*NptgDistrictCodeType*	Optional
AnnotatedNptgLocalityRef			*AnnotatedNptgLocalityRefStructure*	Optional
LocalEducationAuthorityRef			*LocalEducationAuthorityCodeType*	Optional

OrganisationCode is mandated by the schema. There is a recommendation in the schema that this should be the reference number of the school, as allocated by the administrations in the devolved governments, however this is not required by TXC-PTI. Firstly, a textual code can be more meaningful (e.g. COWA = College of West Anglia), and secondly, as has already been noted, a set of dates could cover multiple organisations.

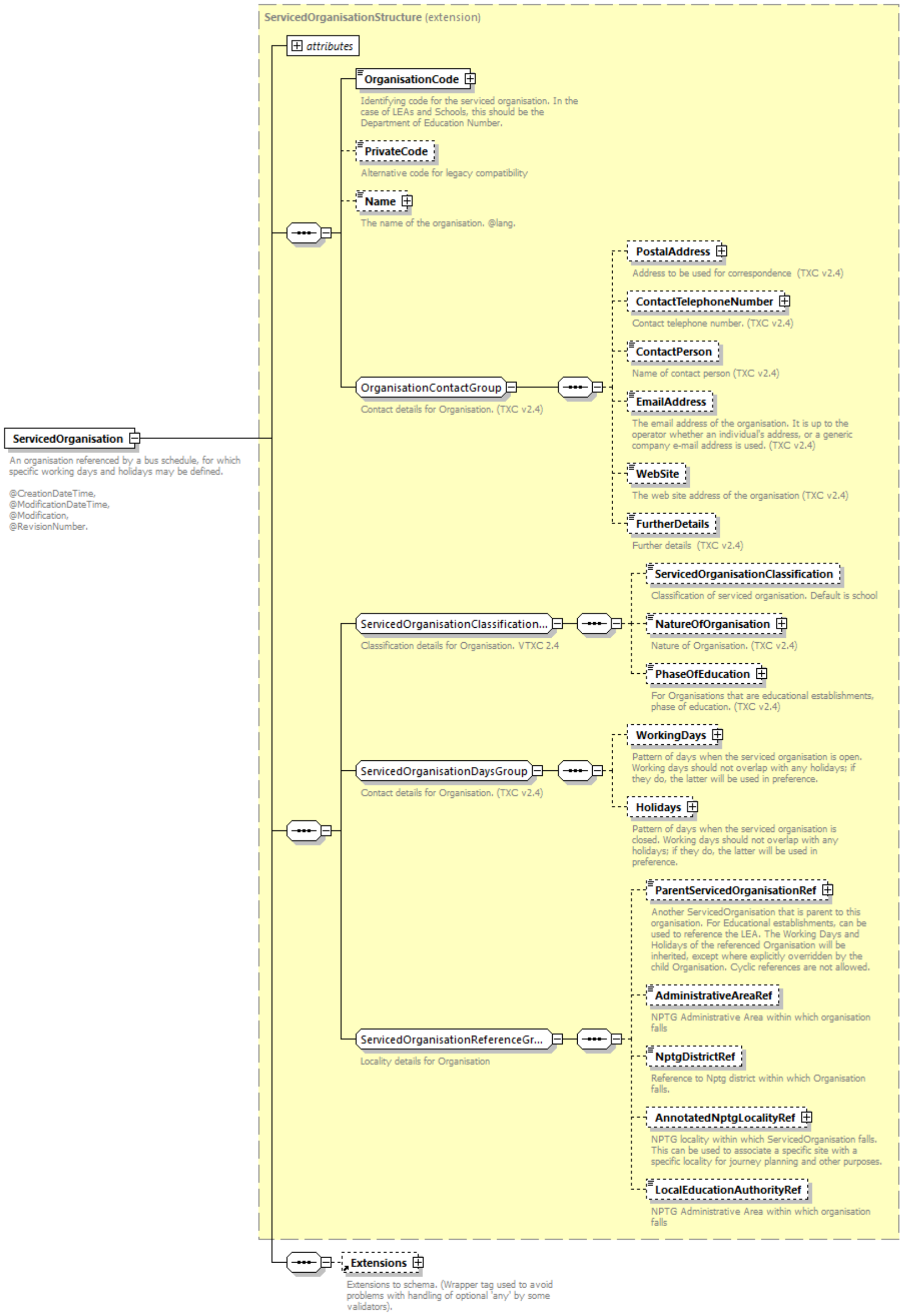

ServicedOrganisationStructure (extension)
attributes
OrganisationCode
Identifying code for the serviced organisation. In the case of LEAs and Schools, this should be the Department of Education Number.
PrivateCode
Alternative code for legacy compatibility
Name
The name of the organisation. @lang.
PostalAddress
Address to be used for correspondence (TXC v2.4)
ContactTelephoneNumber
Contact telephone number. (TXC v2.4)
ContactPerson
Name of contact person (TXC v2.4)
OrganisationContactGroup
Contact details for Organisation. (TXC v2.4)
EmailAddress
The email address of the organisation. It is up to the operator whether an individual's address, or a generic company e-mail address is used. (TXC v2.4)
ServicedOrganisation
An organisation referenced by a bus schedule, for which specific working days and holidays may be defined.
@CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
WebSite
The web site address of the organisation (TXC v2.4)
FurtherDetails
Further details (TXC v2.4)
ServicedOrganisationClassification
Classification of serviced organisation. Default is school
ServicedOrganisationClassification...
Classification details for Organisation. VTXC 2.4
NatureOfOrganisation
Nature of Organisation. (TXC v2.4)
PhaseOfEducation
For Organisations that are educational establishments, phase of education. (TXC v2.4)
WorkingDays
Pattern of days when the serviced organisation is open. Working days should not overlap with any holidays; if they do, the latter will be used in preference.
ServicedOrganisationDaysGroup
Contact details for Organisation. (TXC v2.4)
Holidays
Pattern of days when the serviced organisation is closed. Working days should not overlap with any holidays; if they do, the latter will be used in preference.
ParentServicedOrganisationRef
Another ServicedOrganisation that is parent to this organisation. For Educational establishments, can be used to reference the LEA. The Working Days and Holidays of the referenced Organisation will be inherited, except where explicitly overridden by the child Organisation. Cyclic references are not allowed.
AdministrativeAreaRef
NPTG Administrative Area within which organisation falls
ServicedOrganisationReferenceGr...
Locality details for Organisation
NptgDistrictRef
Reference to Nptg district within which Organisation falls.
AnnotatedNptgLocalityRef
NPTG locality within which ServicedOrganisation falls. This can be used to associate a specific site with a specific locality for journey planning and other purposes.
LocalEducationAuthorityRef
NPTG Administrative Area within which organisation falls
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 7 - The ServicedOrganisation element

Name shall be provided, and shall be a meaningful name for the organisation, or group of organisations, being referenced. For example, “College of West Anglia” or “Staffordshire schools”.

ServicedOrganisationClassification may be provided if additional context is required. This is an enumeration that describes the type of establishment, e.g. “school”, “factory”, etc. However, not all systems will recognise or use this.

WorkingDays provides a series of date ranges when the organisation is open. This date range:

- shall have an inclusive **StartDate**;
- shall have an inclusive **EndDate**;
- may have a **Description** to give some context to the date range, although a description is not mandatory in TXC-PTI; and
- may describe a date range as **Provisional**, although such date ranges must be replaced with actual date ranges prior to the operation of the service.

Note that **Holidays** shall not be provided. Providing both working dates and holiday dates might seem like a benefit, but there have been instances of overlapping date ranges, and strange encodings e.g. where a service is said to not run in the holidays of an organisation. This can lead to confusion and misinterpretation of the data.

In TXC-PTI, therefore, only **WorkingDays** shall be defined, and all other days that are not **WorkingDays** shall be assumed to be holidays. This means that, in a **ServicedOrganisationDayType** within an operating profile, trips which only operate on working days will be encoded as **DaysOfOperation/WorkingDays**, while trips which only operate during the holidays will be encoded as **DaysOfNonOperation/WorkingDays**. It is recommended, however, that any presentation to the user avoids negative descriptions such as “this trip does not operate on the working days of X” but presents it instead with a positive description such as “this trip operates during holidays of X.”

To assist with clarity and simplicity of understating when trips are operating it is suggested that the use of a WorkingDays range for each half term for education establishments is helpful.

See section 9.3 for more details of the **OperatingProfile** element, and in particular the relationship between the working dates and the regular days of operation in section 9.3.6.

In TXC-PTI, where a **ServicedOrganisation** is included, it shall be given a code, a meaningful name, and working dates when the organisation is open. Holiday dates shall not be provided; all non-working dates shall be assumed to be holidays.

Name is required to be at least 5 characters. This is to ensure that meaningful names are used for example “College of West Anglia” or “Staffordshire schools”. Shortenings such as SCH should not be used as they do not make it possible to understand the actual organisation being referred to.

At least one WorkingDays range needs to be provided otherwise the validator will report an error.

When providing Provisional dates if the Startdate for WorkingDays is less than or equal to the current date then the validator will report an error as Provisional dates have to be in the future.

If Holidays is provided, then the validator will report an error.

It is important to ensure continued compliance with the requirement to publish dates of operation for a service. Where a Service uses ServicedOrganisaiton and new or updated WorkingDays are available from the organisation data published on BODS will need to be updated to reflect the latest known WorkingDays.

3.3 Example of a serviced organisation coding in TXC-PTI

The following is an example of how a **ServicedOrganisation** may be coded and used within a TXC-PTI document. For further details about how the data is encoded in an **OperatingProfile**, please see section 9.3.

```
<ServicedOrganisation>
   <OrganisationCode>BurCol</OrganisationCode>
   <Name>Burnley College</Name>
   <WorkingDays>
   <DateRange>
        <StartDate>2019-04-22</StartDate>
        <EndDate>2019-05-24</EndDate>
   </DateRange>
   <DateRange>
        <StartDate>2019-06-03</StartDate>
        <EndDate>2019-06-28</EndDate>
   </DateRange>
   </WorkingDays>
</ServicedOrganisation>
...
<VehicleJourneys>
   <VehicleJourney SequenceNumber="1">
   <PrivateCode>120MFBP03:O:1</PrivateCode>
   <OperatingProfile>
        <RegularDayType>
           <DaysOfWeek>
              <Monday/>
              <Tuesday/>
              <Wednesday/>
              <Thursday/>
              <Friday/>
           </DaysOfWeek>
        </RegularDayType>
        <ServicedOrganisationDayType>
           <DaysOfOperation>
              <WorkingDays>
                 <ServicedOrganisationRef>BurCol</ServicedOrganisationRef>
              </WorkingDays>
           </DaysOfOperation>
        </ServicedOrganisationDayType>
   </OperatingProfile>
   ...
   </VehicleJourney>
   ...
</VehicleJourneys>
```

4 Operators, Garages and Registrations

4.1 Introduction

In a TransXChange document, every trip that operates is operated by an **Operator**. Given that TXC is most often used for bus and coach data, an **Operator** is usually a registered bus or coach operator in possession of a valid, current, OTC-issued “O-licence”. However, it is also possible that an **Operator** is a not-for-profit community transport association with a “Section 22” permit, or an operator of another mode. The schema is agnostic as far as the type of operator is concerned.

4.2 The Operators group

The **Operators** group in TXC is a grouping element that can hold one or multiple operator records, classified into different operating categories: **Operator** or **LicensedOperator**. There is no difference in the content of these two categories; it is simply that **LicensedOperator** contains some additional mandatory elements above what is required for an **Operator**. However, a **LicensedOperator** includes some mandatory information that is better sourced from other sources, and hence only **Operator** shall be used.

The Bus Open Data requirements that will be supported by the TXC-PTI profile are for each operator to be responsible for their own data. Therefore, an **Operators** group shall only contain a single instance of **Operator**.

For TXC-PTI, an **Operators** group shall contain only a single **Operator**, and not a **LicensedOperator**.

If more than one Operator or a LicencedOperator is included then the validator will report an error

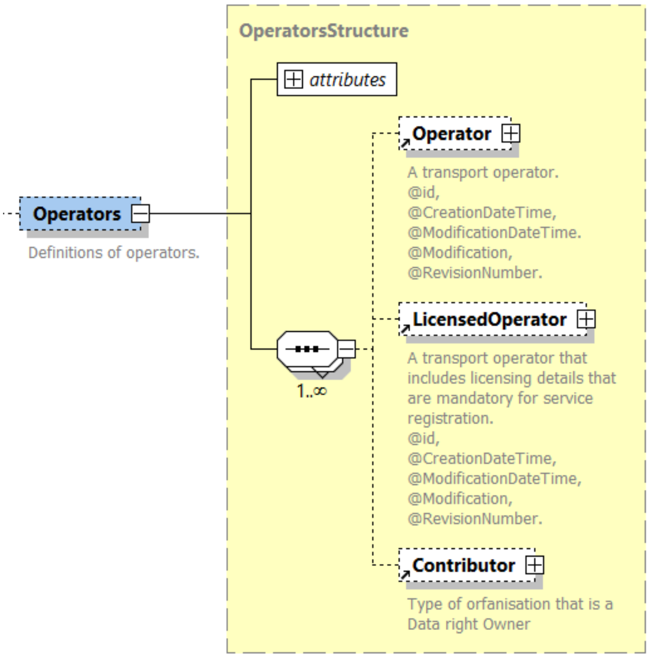


Figure 8 - The Operators structure

4.3 The Operator element

4.3.1 Basic structure of the Operator element

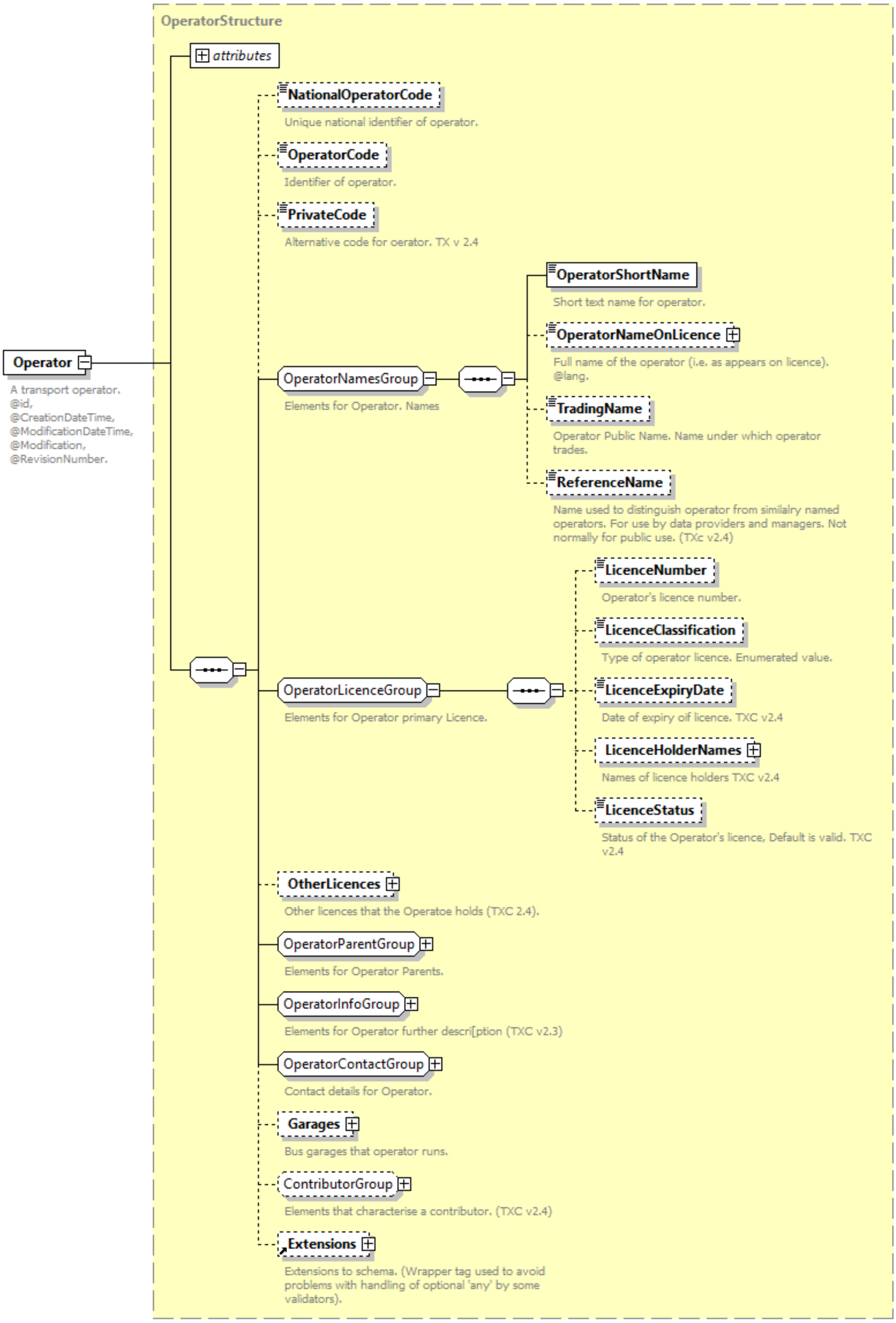

OperatorStructure
attributes
NationalOperatorCode
Unique national identifier of operator.
OperatorCode
Identifier of operator.
PrivateCode
Alternative code for oerator. TX v 2.4
OperatorShortName
Short text name for operator.
OperatorNameOnLicence
Full name of the operator (i.e. as appears on licence). @lang.
Operator
A transport operator. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
OperatorNamesGroup
Elements for Operator. Names
TradingName
Operator Public Name. Name under which operator trades.
ReferenceName
Name used to distinguish operator from similalry named operators. For use by data providers and managers. Not normally for public use. (TXc v2.4)
LicenceNumber
Operator's licence number.
LicenceClassification
Type of operator licence. Enumerated value.
OperatorLicenceGroup
Elements for Operator primary Licence.
LicenceExpiryDate
Date of expiry oif licence. TXC v2.4
LicenceHolderNames
Names of licence holders TXC v2.4
LicenceStatus
Status of the Operator's licence, Default is valid. TXC v2.4
OtherLicences
Other licences that the Operatoe holds (TXC 2.4).
OperatorParentGroup
Elements for Operator Parents.
OperatorInfoGroup
Elements for Operator further descri[ption (TXC v2.3)
OperatorContactGroup
Contact details for Operator.
Garages
Bus garages that operator runs.
ContributorGroup
Elements that characterise a contributor. (TXC v2.4)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 9 - The Operator element

The **Operator** element contains a large amount of information about the operator, most of which is best obtained from other sources by means of referenced identities. There is no need to replicate it within the TXC, and this also avoids the risk that conflicting information is provided; there needs to be a single source of truth. The **Operator** element therefore only needs to contain references which link to those external sources, and such information as is either mandatory in the schema or which cannot necessarily be inferred from external sources.

Each **Operator** shall have an *id* attribute. This is an internal reference only and is only used to reference other elements within the TXC document. The format of this id is not mandated by the TXC-PTI profile, although it is recommended that this be that national operator code, which is both human-readable and meaningful.

The Operator elements shall be used as set out in Table 7.

Table 7 - Permitted elements for Operator

Element Name	**Data Type**	**Used in TXC-PTI**
NationalOperatorCode	*NationalOperatorCodeType*	Yes. Shall be included
OperatorCode	*OperatorCodeType*	Where appropriate
PrivateCode	*PrivateCodeType*	Optional
OperatorShortName	*OperatorShortNameType*	Yes. Mandated by the schema, although limited by the schema to 24 characters
OperatorNameOnLicence	*NaturalLanguageString*	Where appropriate to aid readability / traceability.
TradingName	*OperatorTradingNameType*	Where appropriate
ReferenceName	*normalizedString*	Optional
LicenceNumber	*OperatorLicenceNumberType*	Yes. Shall be included. Licence number specific to the service in this TXC document.
LicenceClassification	*OperatorLicenceEnumeration*	Optional
LicenceExpiryDate	*date*	Optional
LicenceHolderNames	*LicenceHolderNamesStructure*	Optional
LicenceStatus	*LicenceStatusEnumeration*	Optional
Other Licences	*OperatorLicencesStructure*	Optional
ParentOperatorRef	*AnnotatedOperatorRefStructure*	Optional
UltimateParentRef	*AnnotatedOperatorRefStructure*	Optional
PrimaryMode	*VehicleModesEnumeration*	Optional
EbsrUser	*Boolean*	Optional
TravelineOwner	*RegionShortCodeEnumeration*	Optional
RegionalOperatorRefs		Optional
Note	*NaturalLanguageString*	Optional
EnquiryTelephoneNumber	*TelephoneContactStructure*	Optional
ContactTelephoneNumber	*TelephoneContactStructure*	Optional
ContactFaxNumber	*TelephoneContactStructure*	Optional

ContactPerson			*normalizedString*	Optional
CustomerServicesTelephoneNumber			*TelephoneContactStructure*	Optional
OperatorAddresses			*OperatorAddressesStructure*	Optional
EmailAddress			*EmailAddressType*	Optional
WebSite			*anyURI*	Optional
Garages			*GaragesStructure*	Where appropriate to show dead runs for real time information, and not otherwise.
	Garage	GarageCode	*GarageCodeType*	Mandatory if garages are included
		GarageName	*NaturalLanguageString*	Where appropriate
		ContactNumber	*TelephoneContactStructure*	Optional
		Address	*UKPostalAddressStructure*	Optional
		ContactPerson	*normalizedString*	Optional
		Location	*LocationStructure*	Optional
PolicyStatus			*PolicyStatusEnumeration*	Optional
ContributorClassification			*ContributorClassification Enumeration*	Optional
DataRights			*DataRightsStructure*	Not used as all data provided to BODS is governed by the Bus Services Act 2017.

NationalOperatorCode is mandatory. This code shall refer back to the appropriate code held within the National Operator Code database (NOC) maintained and operated by Traveline Information Limited (TIL) or its successor organisations. The publicly accessible NOC database shall hold all the appropriate details for the operator including, but not limited to, contact details, websites, public-facing operator identities, etc. as appropriate. The NOC code shall be used as the key for accessing this information, without it having to be duplicated within the TXC document except as outlined below.

The NationalOperatorCode database is available for download, or to browse at: https://www.travelinedata.org.uk/traveline-open-data/transport-operations/about-2/

It is important to use the correct NOC code for the Service. It is not unusual for an operator to have more than one NOC to reflect different trading names used under the same O-licence. For example:

The Arriva in the South East have an O-Licence as Arriva Kent & Surrey Ltd and have three NOC:

- AKSS Arriva Kent and Surrey
- ASCN Arriva Southern Counties
- GLNK Arriva Southern Counties (GLNK)

Most operators will have a single NOC, for example:

- HULS for Henry Hulley & Sons Ltd using the public name Hulleys of Baslow.
- STEP for Stephenson's Of Easingwold Ltd using the public name Stephensons of Easingwold .

The consistent use of an operators NationalOperatorCode enables data consumers to link different data sets together, for example location data for real time information, or fares for ticketing solutions, more easily than would otherwise be possible.

OperatorCode is optional, although it may be populated for compatibility with other systems

OperatorShortName is mandated by the schema. Where possible this should match the Operator Public Name field in NOC.

TradingName shall be provided, where this is different from the name given in NOC.

LicenceNumber shall be provided where applicable. Where an operator has more than one O-licence, the licence number provided shall be the licence number relevant to the service being described in the TXC document. Details about the licence shall be obtained via reference to OTC, and not to other sources, although **OperatorNameOnLicence** may be provided to aid human readability of the TXC document.

Garage entries in the **Garages** group may be provided, but only inasmuch as they are required for use in e.g. dead runs in **VehicleJourneys** and for use in real time systems. Otherwise, this information shall be left empty.

No other elements shall be populated for **Operator**. This includes, specifically, any contact information contained within the **OperatorContactGroup** as this information shall be taken from NOC referenced by the **NationalOperatorCode**.

For TXC-PTI the operator element shall include **NationalOperatorCode**, **OperatorShortName**, and shall also include **TradingName**, **LicenceNumber** and **Garages** as appropriate. Other elements are optional or not used.

The NationalOperatorCode provided should be the most granular NOC for the operator (e.g. FESX for First Group's bus operations in Essex)

If Garages structure is provided it must include at least one Garage.

4.3.2 Example of an Operator coding

```
<Operator id="EYMS">
   <NationalOperatorCode>EYMS</NationalOperatorCodeCode>
   <OperatorShortName>East Yorkshire Buses</OperatorShortName>
   <OperatorNameOnLicence>East Yorkshire Motor Services</OperatorNameOnLicence>
   <LicenceNumber>PB0000328</LicenceNumber>
   <Garages>
      <Garage>
         <GarageCode>HU</GarageCode>
         <GarageName>Hull</GarageName>
      </Garage>
   </Garages>
</Operator>
```

4.4 Registrations

In EBSR documents, a TXC **Registrations** block is included to replicate the information that would be contained on the paper registration. However, for TXC-PTI which is public-facing information, this information is superfluous and adds nothing to the quality of the open data. For this reason, the **Registrations** block shall not be included in a TXC-PTI document.

There is merit in referencing the OTC registration reference, however, for traceability and completeness purposes, but this shall be encoded as part of the **ServiceCode** within the **Services** element (see 5.2).

➡ The **Registrations** group shall not be included within a TXC-PTI compliant document.

🚌 If a Registrations group is included then the validator will report an error.

5 Services and Lines

5.1 Introduction

In the UK, the label “service” is often used in common terminology but means different things to different people, but in data it is important to make the distinction.

In TXC, this distinction is made using the terms “service” and “line”, where:

- A LINE is a collection of vehicle journeys (trips) that share some high degree of commonality of route and timings, and which are all known by the same LINE NAME. Examples are “1”, “1B”, “100”, “calverton connection”, etc. (there is no requirement for a LINE NAME to include a number)
- A SERVICE is a collection of LINEs that together make up a coherent set of vehicle journeys usually running over geographically similar routes although this isn’t a requirement. For example, consider a service “1/1A”, where line “1” operates “A to B direct” and line “1A” operates “A to B via C”, and together they make up the complete service between A and B.

Note that this description and example should be taken as illustrating the terms and the concept only.

5.2 The Services group

The **Services** group in TXC is a group that holds one or more **Service** elements in a *ServicesStructure*. It has been common practice in some operators’ data to submit several services within the same TXC document. However, there is often little consistency in how these are grouped and, often, multiple services contain single lines where the intent would be better represented by a single service with multiple lines. Additionally, including too much information in a single TXC document can make it harder to fault-find a single service due to the inclusion of data not related to that service. In TXC-PTI, therefore, the permitted number of services in a single TXC document shall be limited.

In TXC-PTI, the **Services** group shall only contain a single **Service** element.

If more than one Service element is provided then the validator will raise an error.

5.3 The Service element

5.3.1 Introduction

The **Service** element in TXC describes the service, the lines that it comprises, and provides other information about the default operation of the service. Certain elements are schema-mandated, however TXC-PTI will provide rules as to the ways that a number of these shall be populated.

As stated earlier, there shall only be one **Service** element in TXC-PTI file. This applies even to different versions of the same service, because a duplication would require different keys to be contained within the data, which would then break the requirement for the keys to be consistent from version to version.

A **Service** shall be versioned, as described in section 2.3 above.

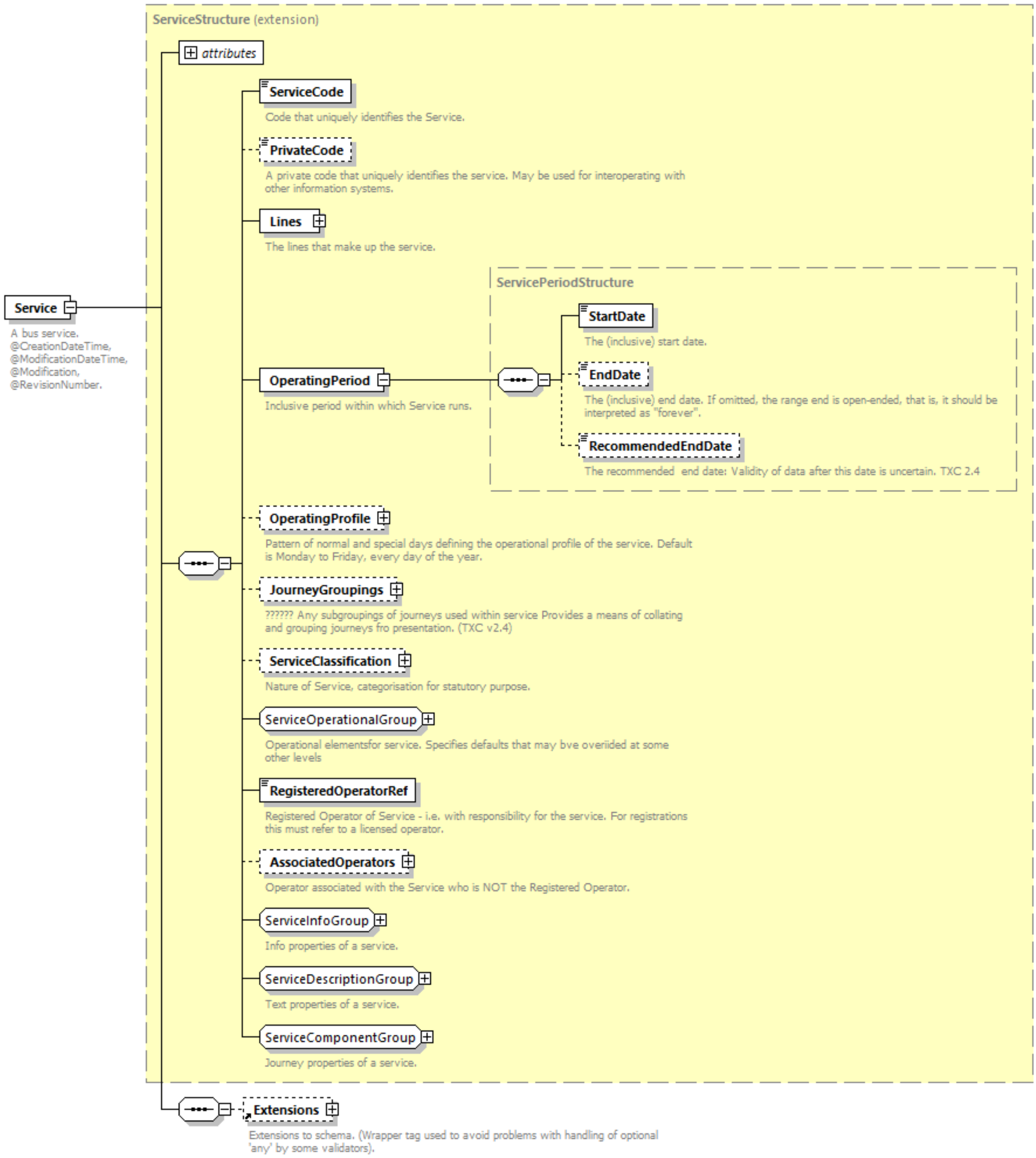

ServiceStructure (extension)
attributes
ServiceCode
Code that uniquely identifies the Service.
PrivateCode
A private code that uniquely identifies the service. May be used for interoperating with other information systems.
Lines
The lines that make up the service.
Service
A bus service. @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
ServicePeriodStructure
StartDate
The (inclusive) start date.
OperatingPeriod
Inclusive period within which Service runs.
EndDate
The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever".
RecommendedEndDate
The recommended end date: Validity of data after this date is uncertain. TXC 2.4
OperatingProfile
Pattern of normal and special days defining the operational profile of the service. Default is Monday to Friday, every day of the year.
JourneyGroupings
?????? Any subgroupings of journeys used within service Provides a means of collating and grouping journeys fro presentation. (TXC v2.4)
ServiceClassification
Nature of Service, categorisation for statutory purpose.
ServiceOperationalGroup
Operational elementsfor service. Specifies defaults that may bve overiided at some other levels
RegisteredOperatorRef
Registered Operator of Service - i.e. with responsibility for the service. For registrations this must refer to a licensed operator.
AssociatedOperators
Operator associated with the Service who is NOT the Registered Operator.
ServiceInfoGroup
Info properties of a service.
ServiceDescriptionGroup
Text properties of a service.
ServiceComponentGroup
Journey properties of a service.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 10 - The Service element

Mandatory, optional and unused elements within a Service are as shown in the table below. Detailed usage of the elements is described more fully in the sections which follow.

Table 8 - Permitted elements for Service

Element Name	Data Type	Used in TXC-PTI
PrivateCode	*PrivateCodeType*	Where appropriate
Lines		Schema mandated. See 5.4.

OperatingPeriod	*ServicePeriodStructure*	Schema mandated. See 0.
OperatingProfile	*OperatingProfileStructure*	Yes, if defining all or the majority of trips in full. Omitted otherwise. See 0.
JourneyGroupings	*JourneyGroupingsStructure*	Not used
ServiceClassification	*ServiceClassificationStructure*	Not used
TicketMachineServiceCode	*TicketMachineServiceCodeType*	Where appropriate to define it at service level rather than individual vehicle journey level. See 0
VehicleType	*VehicleTypeStructure*	Where appropriate. See 5.3.7
RegisteredOperatorRef	*OperatorRefStructure*	Schema mandated. Refers to an operator id defined in the Operator element.
AssociatedOperators	*OperatorRoleStructire*	Where appropriate. See 5.3.4
ServiceHasMirror	*boolean*	Optional
StopRequirements	*StopRequirementsStructure*	Optional
Mode	*VehicleModesEnumeration*	Where appropriate. See 5.3.7
PublicUse	*boolean*	Yes. Shall be included. True if service is available to the public, false if not. See 5.3.7
ServiceAvailability		Optional
Express	*boolean*	Optional
CommercialBasis	*ServiceCommercialBasis Enumeration*	Where appropriate. See 5.3.7
Description	*NaturalLanguageString*	Optional
Note	*NoteStructure*	Where appropriate. See 5.3.7
SchematicMap	*ImageDocumentType*	Not used
MarketingName	*normalizedString*	Where appropriate. See 5.3.7
ToBeMarketedWith	*ToBeMarketedWithStructure*	Where appropriate. See 5.3.4
StandardService	*StandardServiceStructure*	Where appropriate. Shall be included for non-flexible services. See 5.3.5
FlexibleService	*FlexibleServiceStructure*	Where appropriate. Shall be included for flexible services. See 10
Direction	*ServiceDirectionEnumeration*	Not used
JourneyPatternInterchange	*JourneyPatternInterchange Structure*	Where appropriate. See 5.3.6

5.3.2 Identifying the service

5.3.2.1 Registered services

The **ServiceCode** element uniquely identifies the service.

For registered services within TXC-PTI this shall be the registration reference as given when the service is registered with OTC. This takes the form of the "O-licence" number, a forward slash, and a sequential number; for example, PF0000459/134 is the registration reference for Stagecoach service 3 in Bedford. Using the registration reference for the **ServiceCode** in this way facilitates traceability and permits other operations such as completeness checks to be carried out.

While a forward slash is not a reserved character in XML, to avoid confusion and to aid readability the **ServiceCode** in TXC-PTI shall replace the forward slash in the registration reference with a colon.

```
<Service>
   <ServiceCode>PF0000459:134</ServiceCode>
   ...
</Service>
```

Additional examples for ServiceCode:

PF0007157/12 is the registration reference for Kinchbus service 11 and 12 in Leicestershire the service code would be:

```
<Service>
   <ServiceCode>PF0007157:12</ServiceCode>
   ...
</Service>
```

PC2021320/8 is the registration reference for Go North West service 92 and 93 in Manchester, the ServiceCode would be:

```
<Service>
   <ServiceCode>PC2021320:8</ServiceCode>
   ...
</Service>
```

5.3.2.2 Unregistered services

While Bus Open Data is directed initially towards operators of registered bus services, the TXC-PTI profile is intended to fulfil a wider use. Data for *unregistered* services such as community transport services and coach services therefore needs to ensure that **ServiceCode** is a) unique; and b) consistent from version to version in the same way that it is for registered services. Unregistered services, though, do not have an OTC registration reference that can be used.

In TXC-PTI, uniqueness for unregistered services shall be ensured by formatting the **ServiceCode** as follows:

- Start with the letter code "UZ" to denote an unregistered service. The second letter is a placeholder in case a future version of this profile requires to split out unregistered codes e.g. by region in a similar way to OTC's "PF", "PH", etc. O-licence codes.
- Follow with three zeros and then the 4-letter NOC code of the operator (or four zeroes if the NOC code is three letters, or five zeros if the NOC code is two letters).

  These two elements combined provide a format which is the same length and styling as the O-licence for a registered service.

- Follow that with an operator-generated reference number that is unique for that service, and which shall persist from version to version of the same service. Separate this with a colon, as is the case for the registered services.

For example, West Norfolk Community Transport may choose to encode their “Go To Town” service 32 like this:

```
<Service>
  <ServiceCode>UZ000WNCT:GTT32</ServiceCode>
  ...
</Service>
```

Note that the last part of the code is entirely down to the operator. There is no requirement to include the line number, as shown in the example, as the line number or name will be included in the *id* of the Line - see section 5.5.2 below.

To be clear, this method of creating a service code only applies to services which do not have an OTC service registration reference. Where a service has an OTC service registration reference then this shall be used instead, as set out in section 5.3.2.1,even if the service is not legally required to be registered.

In TXC-PTI, where a service has an OTC service registration reference then the **ServiceCode** shall be that service registration reference, with the forward slash replaced with a colon.

Where a service is unregistered, and only then, operators shall use a similar format with the O-licence number being replaced with a pseudo-reference made from “UZ” plus the operator’s NOC and a unique reference as described above.

Data supplied to BODS must be structured in the same manner as the registration to which it relates. For example, where a single public-facing service is formed of multiple registrations, then the service must be split into separate files each with the relevant ServiceCode to align with the registered services. Where there are multiple Lines on the same registration these must all be provided to BODS within the same ServiceCode.

The BODS requires the ServiceCode to be in the structure described to allow data to be referenced to the OTC registrations data as part of the compliance validation process for operators providing data to BODS.

The inclusion of prefixes and suffixes are not allowed.

ServiceCode must remain consistent when supplying updated data to ensure that version management works as described in section 2.3.

If ServiceCode is not of the correct format, then the validator will report an error.

5.3.3 Operating dates and patterns

5.3.3.1 Use in TXC-PTI

In TXC-PTI, the overall dates of operation of the service shall be defined, as with the full TXC schema, by the mandatory **OperatingPeriod** element. As a minimum, an **OperatingPeriod** requires a **StartDate**. Where the last date of operation of the services is known (for example, because the service is terminating in the near future, there is an impending change, or because the service being described only operates for a limited period), then an **EndDate** shall also be provided. If the service is expected to run "until further notice" then the **EndDate** shall be omitted.

In TXC-PTI, the use of an **EndDate** in the far distant future to represent an effective "until further notice" operation shall not be permitted. An **EndDate** shall only describe the actual end date of a service.

Between these (inclusive) start and end dates, the default operation is described by the **OperatingProfile**. In standard TXC documents, this is predominantly used in one of two ways:

a) To describe the overall operation of the service, e.g. "Mondays to Sundays" for a seven-day service, even though different timetables may operate Mondays to Fridays, Saturdays, and Sundays; or

b) To describe the majority of trips e.g. "Mondays to Fridays" even though there are also trips which run on Saturdays and on Sundays.

A principle behind TXC-PTI is that data which is declared as a default and then over-ridden in a more specialised element should be avoided wherever possible, as this leads to confusion. On that basis, it is preferable that the **Service/OperatingProfile** element should be omitted. However, this does mean that the size of files could expand dramatically when there is no need to because all **OperatingProfile** information has to be encoded into individual **VehicleJourney** elements.

For TXC-PTI, therefore, this principle can be relaxed for **OperatingProfile**, so long as the operating profile completely defines the operations of the largest number of trips in full (e.g. it defines Mondays to Fridays excluding Bank Holidays). Individual **VehicleJourney** elements then have either:

a) no **OperatingProfile**, because they follow the default pattern; or

b) a complete set of **OperatingProfile** elements which completely replace the default pattern, and which describe in full when that **VehicleJourney** runs.

Where used in a **Service**, the **OperatingProfile** shall follow the rules described in section 9.3.

In TXC-PTI:

- the **OperatingPeriod** shall only include an **EndDate** if the service has an actual end date.
- The **Service/OperatingProfile** may only describe, in full, the default days of operation and non-operation of the service, which shall be applicable to the majority of the trips. Otherwise, it shall be omitted.
- **Service/OperatingProfile** may not describe the overall operation of the service if the majority of trips do not operate this way.

Where an EndDate it supplied must reflect actual end dates for contracts or known service end date. The longest likely contract for a service is 10 years, the validator therefore checks that an end date is no more than 4026 days (11 years) after the start date.

5.3.3.2 Examples of correct and incorrect use of OperatingProfile

```
<Services>
   <Service>
      <ServiceCode>PA0000123:1</ServiceCode>
      ...
      <OperatingPeriod>
         <StartDate>2020-09-01</StartDate>
      </OperatingPeriod>
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </Service>
</Services>
<VehicleJourneys>
   <VehicleJourney>
      ...
      <!-- No operating profile – all trips run Monday to Friday -->
   </VehicleJourney>
</VehicleJourneys>
```

The XML excerpt above shows a service that is defined as running on all weekdays. There are no **OperatingProfile** elements in the **VehicleJourney** and so the individual trip also runs on all weekdays.

It would also have been valid to omit the **OperatingProfile** from the service altogether and to place it in the **VehicleJourney** only:

```
<Services>
   <Service>
      <ServiceCode>PA0000123:1</ServiceCode>
      ...
      <OperatingPeriod>
         <StartDate>2020-09-01</StartDate>
      </OperatingPeriod>
      ...
   </Service>
</Services>
<VehicleJourneys>
   <VehicleJourney>
      ...
      <!-- All trips run Monday to Friday defined in vehicle journey -->
      <OperatingProfile>
         <RegularDayType>
```

```
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </VehicleJourney>
</VehicleJourneys>
```

The XML excerpt below is valid showing how the majority of trips match the coding in the service, even though there are additional trips on other weekdays. Here, the majority of the trips are Monday to Friday trips, which do not have **OperatingProfile** elements in the **VehicleJourney**, and only the trips which differ have **OperatingProfile** elements

```
<Services>
   <Service>
      <ServiceCode>PA0000123:1</ServiceCode>
      ...
      <OperatingPeriod>
         <StartDate>2020-09-01</StartDate>
      </OperatingPeriod>
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </Service>
</Services>
<VehicleJourneys>
   <VehicleJourney>
      ...
      <!-- No operating profile -trip runs Monday to Friday -->
   </VehicleJourney>
   <VehicleJourney>
      ...
      <!-- No operating profile - trip run Monday to Friday -->
   </VehicleJourney>
   <VehicleJourney>
      ...
      <!—Additional trip on Saturday defined in vehicle journey -->
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Saturday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </VehicleJourney>
</VehicleJourneys>
```

Finally, the following excerpt is not valid since no trips runs Monday to Saturday. This should have been encoded in one of the forms shown above.

```
<Services>
   <Service>
      <ServiceCode>PA0000123:1</ServiceCode>
      ...
      <OperatingPeriod>
         <StartDate>2020-09-01</StartDate>
      </OperatingPeriod>
      <!-- Defines a Monday to Saturday service -->
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
               <Saturday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </Service>
</Services>
<VehicleJourneys>
   <VehicleJourney>
      ...
      <!-- Monday to Friday only defined in vehicle journey -->
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Monday/>
               <Tuesday/>
               <Wednesday/>
               <Thursday/>
               <Friday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </VehicleJourney>
   <VehicleJourney>
      ...
      <!— Saturday only defined in vehicle journey -->
      <OperatingProfile>
         <RegularDayType>
            <DaysOfWeek>
               <Saturday/>
            </DaysOfWeek>
         </RegularDayType>
      </OperatingProfile>
      ...
   </VehicleJourney>
</VehicleJourneys>
```

5.3.4 Service operated by multiple operators

The definition of a service, as stated in the introduction to this section, is that it is a collection of lines that together make up a coherent set of vehicle journeys. The example used in the introduction was of line 1 and line 1A, highlighting one use case of two lines operated by the same operator. However, there is a second class of service, which is one that is operated by multiple operators.

For example, many services are operated as commercial services by operators Monday to Saturday, from early morning through to early evening, with local authorities then contracting a second operator to run evening trips, or trips on Sundays. Another example is a quality partnership, such as is found in Oxford, with two operators alternating trips on the same line to provide a high frequency service to passengers. In both these cases, the two operators' individual services together make up the overall service that the passenger sees.

The Bus Open Data programme requires operators to be responsible for their own data, and the TXC-PTI profile confirms this by only allowing one operator per TXC-PTI document, with reference back to the single-operator service registration and O-licence. It is therefore not possible in TXC-PTI to capture the entirety of a service within a single document when it is operated by different operators.

In order to ensure, as far as possible, that the linkage between the different operators' services is maintained, use should be made within TXC-PTI of the facilities to identify the alternate operators and the alternate services. This is also a suitable mechanism for a single operator to use where there is a split registration with different sections in separate documents based on registrations, and where the operator wishes to indicate that they are linked.

Within **Service**, this linkage can be shown using the **AssociatedOperators** element, which identifies the operator(s) that are linked, and **ToBeMarketedWith**, which identifies the service(s) that should be linked. Together, these give a complete set of information to enable end users to link services, although there is no direct way to link one of a number of associated operators with a specific linked service. The table below highlights the elements that are required and those which are not when including AssociatedOperators and/or ToBeMarketedWith elements.

Table 9 - Usage of AssociatedOperators elements

Element Name	**Data Type**	**Used in TXC-PTI**
OperatorRef	*OperatorCodeType*	A reference to an operator using that operator's NOC
Role	*NaturalLanguageString*	Optional

Table 10 - Usage of ToBeMarketedWith elements

Element Name		**Data Type**	**Used in TXC-PTI**
RelatedService		*AnnotatedServiceRefStructure*	Schema mandated.
	ServiceRef	*ServiceRefStructure*	Optional
	Description	*NaturalLanguageString*	Shall be populated with the service code of the external linked service.

AssociatedOperators contains a list of **OperatorRef** elements. **OperatorRef** is a text entry that, in TXC-PTI, shall be the national operator code of the associated operator.

ToBeMarketedWith contains a list of **RelatedService** elements. A **RelatedService** element itself can, in TXC, contain a **ServiceRef** and/or a **Description**. However, unlike the **OperatorRef** in **AssociatedOperators**, a **ServiceRef** has to refer to a service defined within the same TXC document. For TXC-PTI, therefore, a **RelatedService** shall only contain a Description element which shall be populated with the external service reference (**ServiceCode**) of the linked service.

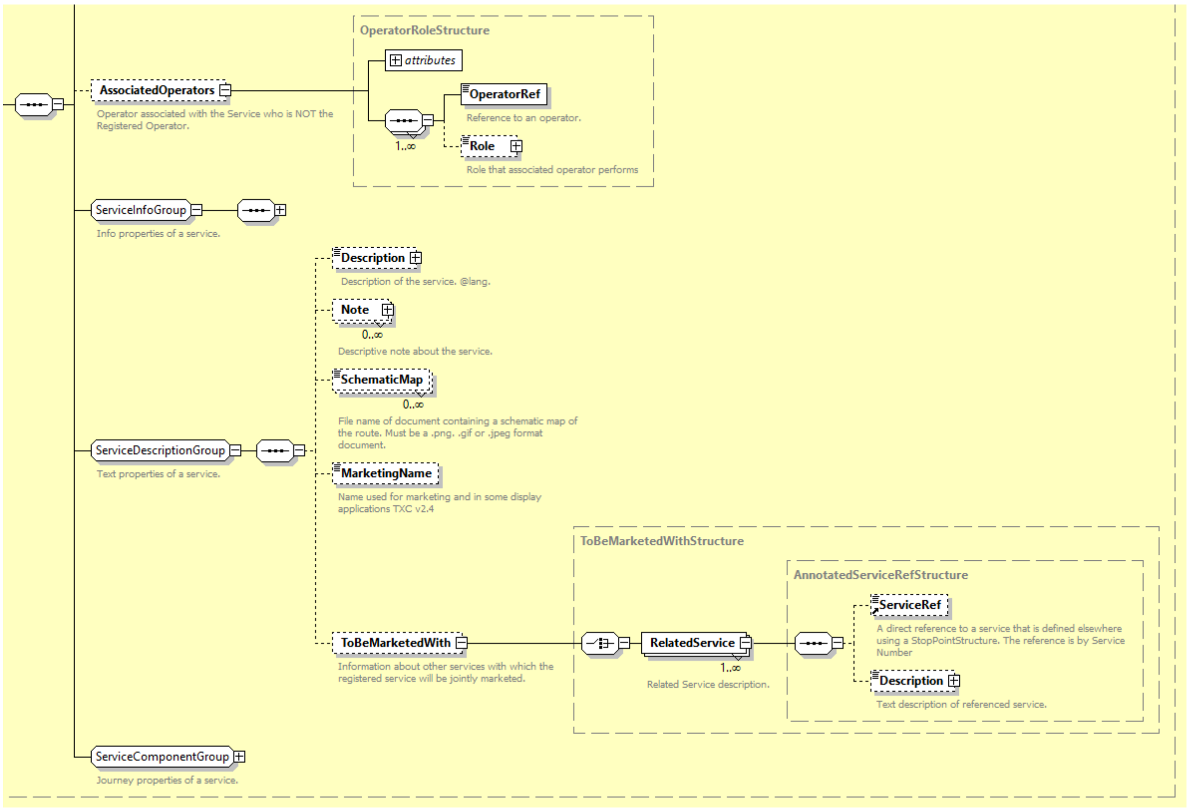

OperatorRoleStructure
attributes
AssociatedOperators
Operator associated with the Service who is NOT the Registered Operator.
OperatorRef
Reference to an operator.
1..∞
Role
Role that associated operator performs
ServiceInfoGroup
Info properties of a service.
Description
Description of the service. @lang.
Note
0..∞
Descriptive note about the service.
SchematicMap
0..∞
File name of document containing a schematic map of the route. Must be a .png. .gif or .jpeg format document.
ServiceDescriptionGroup
Text properties of a service.
MarketingName
Name used for marketing and in some display applications TXC v2.4
ToBeMarketedWithStructure
AnnotatedServiceRefStructure
ServiceRef
A direct reference to a service that is defined elsewhere using a StopPointStructure. The reference is by Service Number
ToBeMarketedWith
Information about other services with which the registered service will be jointly marketed.
RelatedService
1..∞
Related Service description.
Description
Text description of referenced service.
ServiceComponentGroup
Journey properties of a service.

Figure 11 - Excerpt from Service element showing association elements

For example, the code snippet below shows a service which is related to one operated by the operator with a national operator code of “PQRS”, and specifically refers to a service with service code PH0001045:22. It is the responsibility of the downstream user to identify this service and to fetch the data for it.

```
<AssociatedOperators>
   <OperatorRef>PQRS</OperatorRef>
</AssociatedOperators>
...
<ToBeMarketedWith>
   <RelatedService>
      <Description>PH0001045:22</Description>
   </RelatedService>
</ToBeMarketedWith>
```

In TXC-PTI, linked services from different operators may not appear in a single TXC document.

In TXC-PTI, services which are operated by multiple operators may be linked using **AssociatedOperators** and **ToBeMarketedWith**. The **OperatorRef** shall be the NOC of the linked operator, and the **Description** for the **RelatedService** shall contain the **ServiceCode** of the linked service.

This mechanism may also be used to link different documents (services) from the same operator where there is a split registration.

5.3.5 Standard services

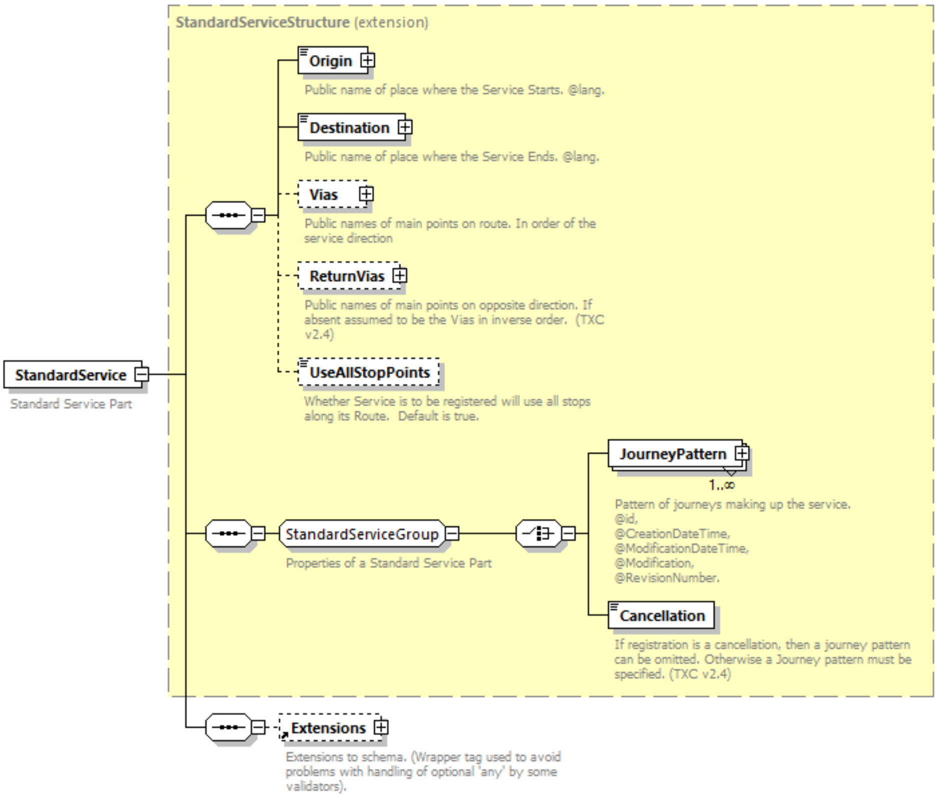

StandardServiceStructure (extension)
Origin
Public name of place where the Service Starts. @lang.
Destination
Public name of place where the Service Ends. @lang.
Vias
Public names of main points on route. In order of the service direction
ReturnVias
Public names of main points on opposite direction. If absent assumed to be the Vias in inverse order. (TXC v2.4)
UseAllStopPoints
Whether Service is to be registered will use all stops along its Route. Default is true.
StandardService
Standard Service Part
JourneyPattern
1..∞
Pattern of journeys making up the service. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
StandardServiceGroup
Properties of a Standard Service Part
Cancellation
If registration is a cancellation, then a journey pattern can be omitted. Otherwise a Journey pattern must be specified. (TXC v2.4)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 12 -The StandardService element

Within the Service element, TXC can describe the default operating pattern of the service, either as a standard service or as a flexible service. A standard service is one which follows a set of stops, in sequence, to a strict timetable, while a flexible service is one which may or may not follow a set of stops in sequence and operates flexibly as and when it is booked. It may therefore not run at all and may only run to some of the stops that are declared.

TXC-PTI is only concerned at present with standard services, however flexible services are discussed briefly in section 10 for future extensions of the profile.

In TXC-PTI, a **StandardService** element is required for standard, non-flexible services, because there needs to be at least one **JourneyPattern** defined and referenced in a **VehicleJourney**.This **StandardService** shall have elements as set out in the table below.

Table 11 - Permitted elements for StandardService

Element Name	Data Type	Used in TXC-PTI
Origin	*NaturalLanguageString*	Schema mandated.
Destination	*NaturalLanguageString*	Schema mandated.
Vias	*ViasStructure*	Where appropriate but avoid use where possible.
ReturnVias	*ViasStructure*	Where appropriate but avoid use where possible.
UseAllStopPoints	*boolean*	Not used
JourneyPattern	*JourneyPatternStructure*	Yes. Shall include at least one **JourneyPattern**. See 8.2
Cancellation	*EmptyType*	Not used (schema mandated to be omitted if **JourneyPattern** present)

Origin and **Destination** shall be included, as these are mandated by the schema.

Vias and **ReturnVias** are optional, however these are better completed for the individual lines and hence can largely be ignored for TXC-PTI.

The **UseAllStopPoints** flag is not to be used, as this is not public facing information and all the stopping points that the service uses should be encoded into the individual vehicle journeys via references to journey patterns.

The **StandardService** shall include one or more **JourneyPattern** elements sufficient to describe at least one pattern in each direction. For details of how a **JourneyPattern** is to be structured, refer to section 8.2 below.

In TXC-PTI, a non-flexible service shall have a **StandardService** which shall consist of at least one **JourneyPattern**.

At least one JourneyPattern must be provided for a StandardService otherwise the validator will report an error.

5.3.6 Interchanges

5.3.6.1 Introduction

Interchanges in TXC can be problematic, because all of the trips that are required for interchange need to be in the same TXC document. This means that, in many cases and particularly for split registrations, it is not easy to describe the interchanges. Other methods of communicating this data have been therefore been developed, although these show some supplier variation and are not part of a standardised TXC exchange.

However, this does not mean that TXC is incapable of describing interchanges, or that TXC-PTI documents should not try to describe interchanges when all of the required information is already in the document. This is particularly true for circular and “pan-handle” services where the outbound trips merge seamlessly into the inbound trips (and/or *vice versa*) and where passengers can remain in the vehicle. Here, the relevant trips would all be in the same document, since this is a single line.

A secondary class of interchange of this type is where a line name changes to another line name using the same vehicle, and again passengers are allowed to remain on the vehicle, or where there is a guaranteed connection between two lines albeit that passengers have to change (e.g. the vehicles “kiss and turn”). These secondary usages are more limited in scope, because the two lines would require to be in the same document and that cannot be guaranteed.

However, wherever interchanges can be defined, it is recommended that a TXC-PTI document should provide this information using TXC’s **JourneyPatternInterchange** element.

It is recommended that, in TXC-PTI, interchanges which can be made on the same service (including between different lines on the same service that are included in the same TXC-PTI document) should be included using **JourneyPatternInterchange** elements as described below.

The **JourneyPatternInterchange** element is the way that interchanges are described at a service level in a TXC file. This describes interchanges that can always be made, regardless of the trip operating. Where possible connections exist that vary from trip to trip (for example, only some journeys are through journeys) then this should be described by a **VehicleJourneyInterchange** instead (see section 9.6).

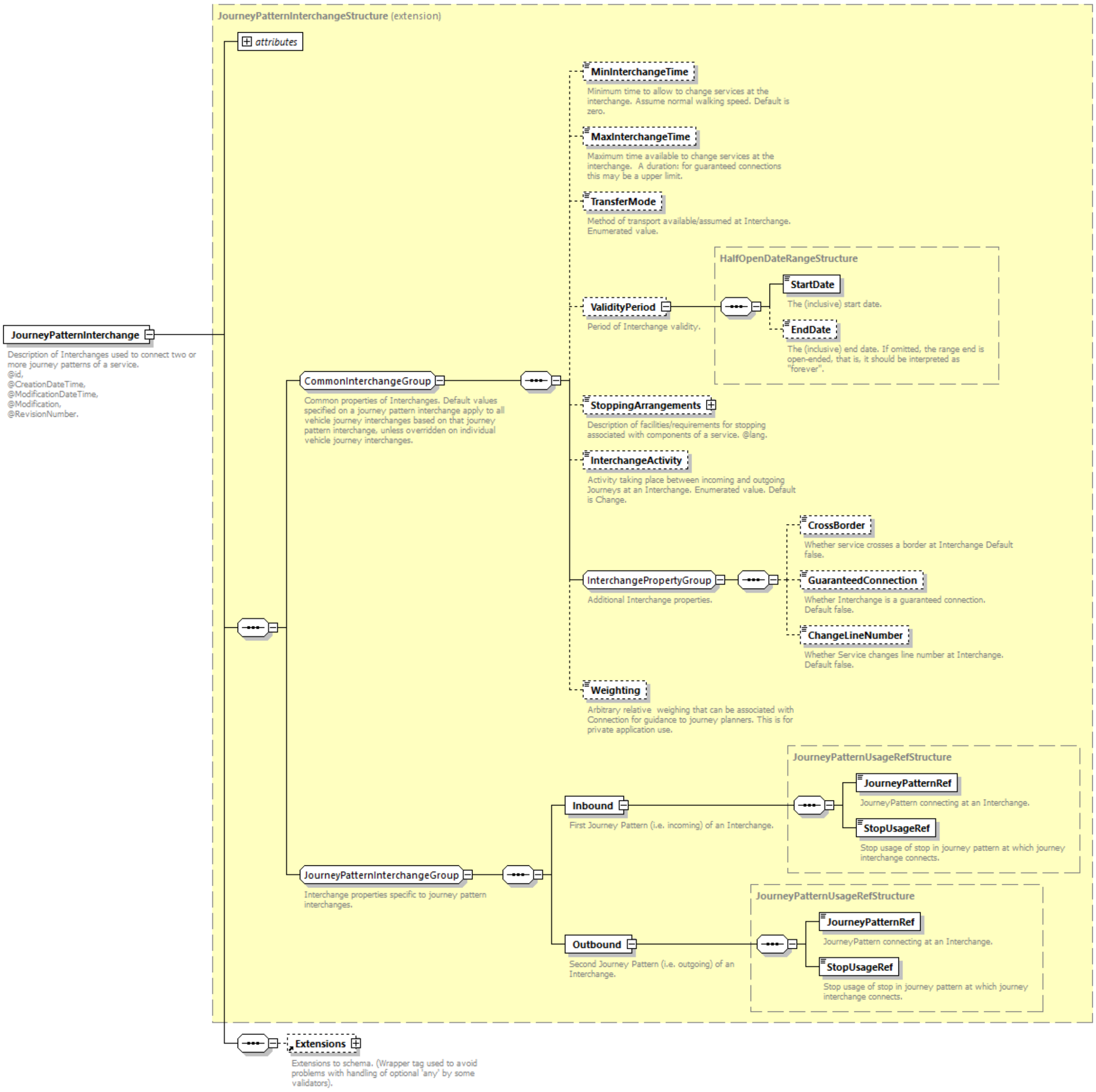

JourneyPatternInterchangeStructure (extension)
attributes
MinInterchangeTime
Minimum time to allow to change services at the interchange. Assume normal walking speed. Default is zero.
MaxInterchangeTime
Maximum time available to change services at the interchange. A duration: for guaranteed connections this may be a upper limit.
TransferMode
Method of transport available/assumed at Interchange. Enumerated value.
HalfOpenDateRangeStructure
StartDate
The (inclusive) start date.
ValidityPeriod
Period of Interchange validity.
EndDate
The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever".
JourneyPatternInterchange
Description of Interchanges used to connect two or more journey patterns of a service. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
CommonInterchangeGroup
Common properties of Interchanges. Default values specified on a journey pattern interchange apply to all vehicle journey interchanges based on that journey pattern interchange, unless overridden on individual vehicle journey interchanges.
StoppingArrangements
Description of facilities/requirements for stopping associated with components of a service. @lang.
InterchangeActivity
Activity taking place between incoming and outgoing Journeys at an Interchange. Enumerated value. Default is Change.
CrossBorder
Whether service crosses a border at Interchange Default false.
InterchangePropertyGroup
Additional Interchange properties.
GuaranteedConnection
Whether Interchange is a guaranteed connection. Default false.
ChangeLineNumber
Whether Service changes line number at Interchange. Default false.
Weighting
Arbitrary relative weighing that can be associated with Connection for guidance to journey planners. This is for private application use.
JourneyPatternUsageRefStructure
JourneyPatternRef
Inbound
JourneyPattern connecting at an Interchange.
First Journey Pattern (i.e. incoming) of an Interchange.
StopUsageRef
Stop usage of stop in journey pattern at which journey interchange connects.
JourneyPatternInterchangeGroup
Interchange properties specific to journey pattern interchanges.
JourneyPatternUsageRefStructure
JourneyPatternRef
JourneyPattern connecting at an Interchange.
Outbound
Second Journey Pattern (i.e. outgoing) of an Interchange.
StopUsageRef
Stop usage of stop in journey pattern at which journey interchange connects.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 13 - The JourneyPatternInterchange element

A **JourneyPatternInterchange** is made up of two groups of elements. The first of these is the common interchange group, which is a set of elements that are common to both journey pattern interchanges and vehicle journey interchanges, and the second of these is the journey pattern interchange group, which is a set of elements that are particular to a journey pattern interchange.

Where a TXC-PTI document chooses to have default journey patterns with zero-minute link durations and actual times in vehicle journeys (see section 8.1), a **JourneyPatternInterchange** shall not be used and a **VehicleJourneyInterchange** shall be used instead. **VehicleJourneyInterchange** elements are described in section 9.4.

5.3.6.2 The common interchange group

The common interchange group consists of a number of elements, with permitted values in TXC-PTI as shown in Table 12 below.

Table 12 - Permitted elements in the common interchange group

Element Name		**Data Type**	**Use in TXC-PTI**
MinInterchangeTime		*DurationType*	The minimum permitted interchange time (default = 0 mins). Use where appropriate
MaxInterchangeTime		*DurationType*	The maximum time for a guaranteed connection. Use where appropriate.
TransferMode		*AllModesEnumeration*	Not used as assumed that the interchange mode is the mode of the service.
ValidityPeriod	StartDate	*StartDateType*	Not used as assumed that the interchange is valid for the duration of the **OperatingPeriod**.
	EndDate	*EndDateType*	
StoppingArrangements		*NaturalLanguageString*	Not used.
InterchangeActivity		*InterchangeActivity Enumeration*	Mandated in TXC-PTI interchanges. Describes the interchange as follows: *transferOnly* not used. *change* used for change of vehicle. *through* used to stay on vehicle. *split* not used. *join* not used.
CrossBorder		*boolean*	Optional.
GuaranteedConnection		*boolean*	Mandated in TXC-PTI interchanges. Set to true for a guaranteed connection which require changes of vehicle, or for through connections with the same line name. False otherwise.
ChangeLineNumber		*boolean*	Mandated in TXC-PTI interchanges. Set to true where a through connection changes line name. False otherwise.
Weighting		*integer*	Optional.

The valid InterchangeActivity enumerations for BODS are *change* and *through* only.

5.3.6.3 The journey pattern interchange group

The specific details of the journey pattern interchange are defined within the journey pattern interchange group. There are two elements, **Inbound** and **Outbound**, both of which are mandated by the schema. Each of these is constructed similarly, in that there is a reference to a journey pattern (**JourneyPatternRef**), and the NaPTAN reference (Atcocode) for the stop point at which the interchange takes place in that journey pattern (**StopUsageRef**). Both **JourneyPatternRef** and **StopUsageRef** are mandatory in the schema.

Inbound shall be used for the feeder (incoming) journey pattern, and **Outbound** shall be used for the distributor (outgoing) journey pattern. Note that the terminology here is slightly confusing, in that inbound and outbound refer to trips incoming and outgoing from the interchange point and do not necessarily match with the inbound and outbound direction of the trip within the line/service.

5.3.6.4 Examples of a journey pattern interchange

The following examples show how an interchange might be defined at a service level for First in Leicester lines 16, a “pan handle” service, and 17, a circular.

The pan handle line 16 has connections between two different journey patterns in different directions (although reference would need to be made to the journey patterns in question to confirm that this was actually the case). The connection is made at Atcocode 269039050, and for a panhandle service it is clear that this needs to be the same stop on the feeder as the distributor.

```
<JourneyPatternInterchange>
   <MinInterchangeTime>PT0M</MinInterchangeTime>
   <InterchangeActivity>through</InterchangeActivity>
   <GuaranteedConnection>true</GuaranteedConnection>
   <Inbound>
      <JourneyPatternRef>JP_12-16-_-y08-1-1-H-1</JourneyPatternRef>
      <StopUsageRef>269039050</StopUsageRef>
   </Inbound>
   <Outbound>
      <JourneyPatternRef>JP_12-16-_-y08-1-5-R-2</JourneyPatternRef>
      <StopUsageRef>269039050</StopUsageRef>
   </Outbound>
</JourneyPatternInterchange>
```

The circular line 17 has connections between the same journey pattern in the same direction (although again this is not clear without reference back to the journey pattern definitions). As can be seen, it is identical in construction to the pan handle interchange.

```
<JourneyPatternInterchange>
   <MinInterchangeTime>PT0M</MinInterchangeTime>
   <InterchangeActivity>through</InterchangeActivity>
   <GuaranteedConnection>true</GuaranteedConnection>
   <Inbound>
      <JourneyPatternRef>JP_12-17-_-y08-1-1-H-1</JourneyPatternRef>
      <StopUsageRef>269057024</StopUsageRef>
   </Inbound>
   <Outbound>
      <JourneyPatternRef>JP_12-17-_-y08-1-1-H-1</JourneyPatternRef>
      <StopUsageRef>269057024</StopUsageRef>
   </Outbound>
</JourneyPatternInterchange>
```

As routes (short workings, etc.), timings, and hence journey patterns, can change across the course of the day, or days of the week, there will of necessity be a number of different combinations of interchanges to suit the differing journey pattern combinations.

5.3.7 Other elements

Other elements in **Service** shall be populated as shown below. Refer back to Table 8 to see their place in the overall Service element and confirmation of their usage.

- **RegisteredOperatorRef** shall be populated, as per the schema, with the (internal) reference of the operator defined in the **Operator** block.
- Where a particular vehicle type with the same set of accessibility features, is commonly used on all trips, **VehicleType** may be populated at **Service** level. This can then be omitted in all **VehicleJourney** elements except those where it is known

to be different. Where included, **VehicleType** shall be populated as set out in 2.4.2 above. As indicated therein, this only indicates the intent of the operator, and is not a guarantee of any particular vehicle actually operating the service as this may change for operational reasons at very short notice.

- **Mode** shall be populated at **Service** level on the same basis.
- **PublicUse** shall be included and shall state explicitly whether a service is available to the public (true) or whether it is a closed service (false). This allows school services, etc. which are closed to nevertheless be entered into data, and hence allow trips to school to be planned for students. This flag shall be in place of the **SchoolOrWorks** designation, part of the **ServiceClassification**. **ServiceClassification** is related to registrations and is not public-facing, and hence shall not be used within a TXC-PTI document.

  Note that a service may be open to the public but have restricted boarding / alighting. For example, the last stop on a public service may be within school grounds first thing in the morning, and only students would be allowed to use it. This restriction is derived from NaPTAN, where a stop may be flagged as private, and users need to reflect both the nature of the service and the nature of the stops when presenting information to the public.
- **CommercialBasis** is a flag that denotes whether or not the service is operated commercially (notContracted) or whether it is wholly or partially contracted (contracted, partContracted). While this is of limited use to the public, contracting authorities are keen to demonstrate their role in providing socially necessary services. Hence this flag is optional, but may be provided at **Service** level if required. At this level, it denotes that all trips have this commercial basis unless indicated otherwise.
- One or more **Note** elements may be provided, where appropriate. As far as possible, however, all information that is in a note which is encodable as data within a TXC-PTI document should be so encoded. Notes shall be as described in section 2.5.
- **MarketingName** may be included where appropriate. Where a name is simply a long service name, this shall be included in the **Line** element instead. For example, Trent Barton's "black cat" service shall be given that **LineName**. However, the First Norfolk & Suffolk "excel" is a marketing name for a group of lines with names "A", "B", "C", etc.
- **PrivateCode** is optional and may be populated if required by legacy systems.

All other elements not explicitly discussed above shall be omitted in a TXC-PTI document. In particular, this includes the **Description** and **Direction** elements:

- **Description** element was intended to replicate the description box on a registration document, where an operator would describe the change to the service in this registration e.g. "Re-timed journeys in the morning to improve reliability". As TXC has become more widespread, this has often been repurposed to give an overall description of the service e.g. "Hereford – Worcester via Ledbury and Great Malvern". This, though, is unidirectional and there are better ways of achieving the same end as described in 5.5.4 below.
- **Direction** is intended to define the default direction of the service. However, this can lead to confusion and in TXC-PTI the direction will be stated explicitly on each journey patterns.

In TXC-PTI, **VehicleType**, **Mode**, **Note**, **MarketingName**, **PrivateCode** and **CommercialBasis** may be included where appropriate. Where they are, they shall be structured as set out above.

In TXC-PTI, **RegisteredOperatorRef** and **PublicUse** shall be populated. With the exception of the optional elements listed above and those described in earlier sections, no other elements within **Service** shall be populated (see Table 8).

5.4 The Lines group

The **Lines** group in TXC is a grouping element within a **Service** that can hold one or multiple line records. As noted above a service may consist of one or more related lines, and consequently a **Lines** group in TXC-PTI may also hold one or multiple line records.

As well as the use case outlined in 5.1 above, an operator may also operate two distinct patterns of operation which are seasonal in nature, i.e. a summer and winter timetable. Where these do not vary from year to year, the operator would not wish to continually update the data to swap between the two and would wish instead to have both patterns within the same TXC-PTI document.

This type of arrangement could very easily be handled through individual dated **VehicleJourney** records. However, this doesn't give a clear separation of the timetables and analysis of the individual records would need to be carried out to understand how to present the "summer" and "winter" timetables to the public.

For these situations, then, it is preferable to have the two patterns coded as distinct lines, albeit sharing the same **LineName**, within TXC-PTI. It is then clear through the referencing which pattern is which, and which trips belong with which pattern.

For TXC-PTI, a **Lines** group may contain multiple **Line** elements, so long as the lines are related either spatially (e.g. 1/1A) or temporally (e.g. summer & winter timetables on line 1).

Where multiple Line elements are provided then each Line must have at least two StopPoints in common with other lines in the same TransXChange file.

5.5 The Line element

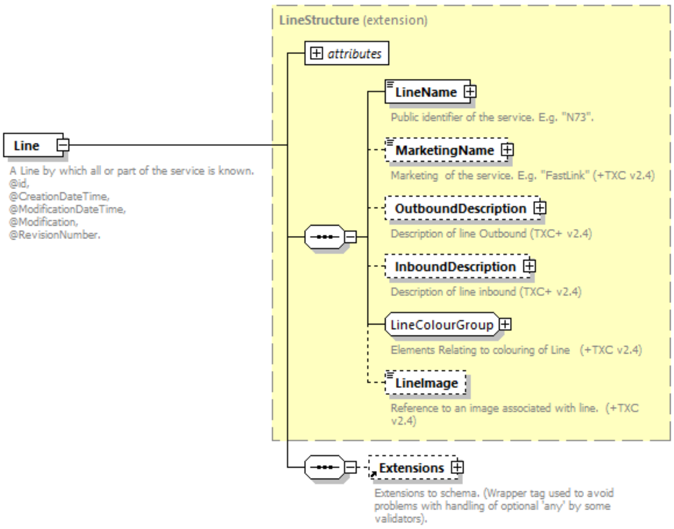

LineStructure (extension)
attributes
LineName
Public identifier of the service. E.g. "N73".
Line
A Line by which all or part of the service is known.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
MarketingName
Marketing of the service. E.g. "FastLink" (+TXC v2.4)
OutboundDescription
Description of line Outbound (TXC+ v2.4)
InboundDescription
Description of line inbound (TXC+ v2.4)
LineColourGroup
Elements Relating to colouring of Line (+TXC v2.4)
LineImage
Reference to an image associated with line. (+TXC v2.4)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 14 - The Line element

5.5.1 Introduction

The **Line** element in TXC describes one of the lines that comprise the service, provides a unique identity for the line, and describes how the line shall be known to the public. It also specifies the separate inbound and outbound descriptions of the service, e.g. outbound as “Hereford to Worcester via Ledbury and Great Malvern”, inbound as “Worcester to Hereford via Great Malvern and Ledbury”.

As stated earlier, there may be multiple **Line** records in a single TXC-PTI document. However, there shall only ever be one version of a given line in a single document. This is true even for seasonal timetables – the two lines represent a single version which encompasses two distinct operating patterns.

A **Line** shall be versioned, as described in 2.3 above.

Mandatory, optional and unused elements within a **Line** are as shown in the table below. Detailed usage of the elements is described more fully in the sections which follow.

Table 13 - Permitted elements for Line

Element Name	**Data Type**	**Used in TXC-PTI**
LineName	*NaturalLanguageString*	Mandatory in schema. See this section.
MarketingName	*NaturalLanguageString*	Where appropriate. See section 5.5.3
OutboundDescription	*LineDescriptionStructure*	Where appropriate, although must have one of OutboundDescription or InboundDescription. See 5.5.4
InboundDescription	*LineDescriptionStructure*	Where appropriate, although must have one of OutboundDescription or InboundDescription. See 5.5.4
LineColour	*ColourValueType*	Where appropriate.

LineFontColour	*ColourValueType*	Optional
AlternativeLineColour	*ColourValueType*	Optional
AlternativeLineFontColour	*ColourValueType*	Optional
LineImage	*anyURI*	Optional

5.5.2 Identifying the line

The *id* attribute of the **Line** element uniquely identifies the line. For TXC-PTI this shall take the form of a number of different components, separated by colons, as follows:

- The most granular NOC for the operator (e.g. FESX for First Group's bus operations in Essex);
- The **ServiceCode** (to assist in linking to other lines in the same group of lines); and
- The line name.

For the avoidance of doubt, the line *id* shall not include a version number. Version control shall be in accordance with 2.3 above.

For example, two distinct line 8s operated by First in Essex might appear (in separate TXC-PTI documents, within their own individual **Service** elements) as:

```
<Service RevisionNumber="1">
   <ServiceCode>PF0000599:41</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:41:8">
         <LineName>8</LineName>
      </Line>
   <Lines>
   ...
   <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
   ...
</Services>
```

and

```
<Service RevisionNumber="4">
   <ServiceCode>PF0000599:75</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:75:8">
         <LineName>8</LineName>
      </Line>
   <Lines>
   ...
   <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
   ...
</Services>
```

Where a service contains two lines which are the seasonal variants of the same service, this can be indicated by the addition of an additional seasonal identifier at the end of the line *id*. This should be separated from the rest of the id by a colon.

For example, if the service 8 in the example above had summer and winter timetables, then these could be identified as follows:

```
<Service RevisionNumber="4">
   <ServiceCode>PF0000599:75</ServiceCode>
   <Lines>
      <Line id="FESX:PF0000599:75:8:winter">
         <LineName>8</LineName>
      </Line>
```

```
    <Line id=" FESX:PF0000599:75:8:summer">
      <LineName>8</LineName>
    </Line>
  <Lines>
  ...
  <RegisteredOperatorRef>FESX</RegisteredOperatorRef>
  ...
</Services>
```

Note that some services may consist of lines that operate all year round, with additional summer trips on a different line number. In this case it is acceptable in TXC-PTI to place the seasonal identifier on the end of the seasonal line *id*, even though there is no corresponding winter timetable.

Note also that such timetables will, in any event, need to be refreshed at least once a year in order to put the correct dates of operation for that year into the service.

Seasonal timetables are the only instance in TXC-PTI where a line may be split. It is not permitted, for example, to have separate TXC-PTI lines or documents for e.g. Mondays to Fridays, Saturdays, etc. operations.

In TXC-PTI, the line *id* shall follow a consistent format of the NOC, the **ServiceCode**, the line name, and, if appropriate, the seasonal identifier, separated by colons. It shall not include a version number.

5.5.3 Line name, line colour and marketing name

The **LineName** element is mandatory and shall contain the public-facing name of the service, in full. For example, Trent Barton's "indigo" service shall have a **LineName** of "indigo", and not any shortened version of it (e.g. "IGO").

Where information about a Line is split across registrations and is therefore contained in multiple Services the LineName shall be consistent across each Service.

The LineColour element is optional and shall be included when a line has a preferred colour to use when showing line in graphic media.

The **MarketingName** is optional and shall be included where a line has a marketing name that is different to other lines within the service. Otherwise, **MarketingName** shall be omitted and any marketing name included at the **Service** level.

5.5.4 Describing the line

In a TXC-PTI document, the line shall have a description for each direction of the line that describes the overall route of the line in general terms. It does not have to describe every place served but should be sufficient to identify the line in public-facing publicity.

This is achieved by use of a line's **OutboundDescription** and **InboundDescription** elements. Both of these are constructed the same way, using a LineDescriptionStructure as shown below:

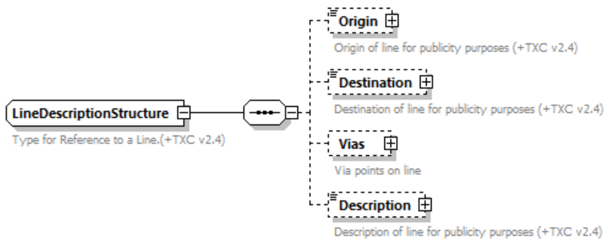

LineDescriptionStructure
Type for Reference to a Line.(+TXC v2.4)
Origin
Origin of line for publicity purposes (+TXC v2.4)
Destination
Destination of line for publicity purposes (+TXC v2.4)
Vias
Via points on line
Description
Description of line for publicity purposes (+TXC v2.4)

Figure 15 - The line description structure

These elements shall be completed as per the table below.

Table 14 - Permitted elements in a LineDescriptionStructure

Element Name	**Data Type**	**Used in TXC-PTI**
Origin	*NaturalLanguageString*	Where appropriate
Destination	*NaturalLanguageString*	Where appropriate
Vias		Where appropriate but avoid use where possible..
Description	*NaturalLanguageString*	Yes. Shall be included

This structure allows the description to be specified granularly, using **Origin**, **Destination**, and a set of some or all of the **Vias** (listed in order of calling), and these elements can be populated in TXC-PTI if required. The **Description** field, however, is mandatory and shall contain a pre-formatted textual description of the inbound and outbound directions. The earlier example showed, for instance, that a line might be described as “Hereford to Worcester via Ledbury and Great Malvern” outbound, and as “Worcester to Hereford via Great Malvern and Ledbury” inbound.

Some general points to note, for TXC-PTI:

- For each direction of the line, there shall be a corresponding description.
- If any direction is not present in the line, the corresponding description shall be omitted.
- There are additional directions that are possible for a line, including *clockwise* and *anti-clockwise*. For the purposed of TXC-PTI, the **OutboundDescription** shall be used for *clockwise* directions, and the **InboundDescription** shall be used for *anti-clockwise* directions.

In TXC-PTI, each direction of a line shall have a matching description element that describes the route of that particular line in that direction. Clockwise and anti-clockwise services are mapped to outbound and inbound description elements respectively.

The validator requires OutboundDescription to be provided unless InboundDescription is provided and vice versa.

6 Stop points and stop areas

6.1 Introduction

In a TXC document, stop point declarations are required so that they can be referenced, ultimately, in the individual vehicle journeys. There are two ways provided in TXC for doing this:

a) by referencing a stop already in NaPTAN; or

b) by creating a new stop record in the same format as a stop record in NaPTAN.

In addition, TXC allows the creation of NaPTAN “stop areas” to replace or supplement those in NaPTAN.

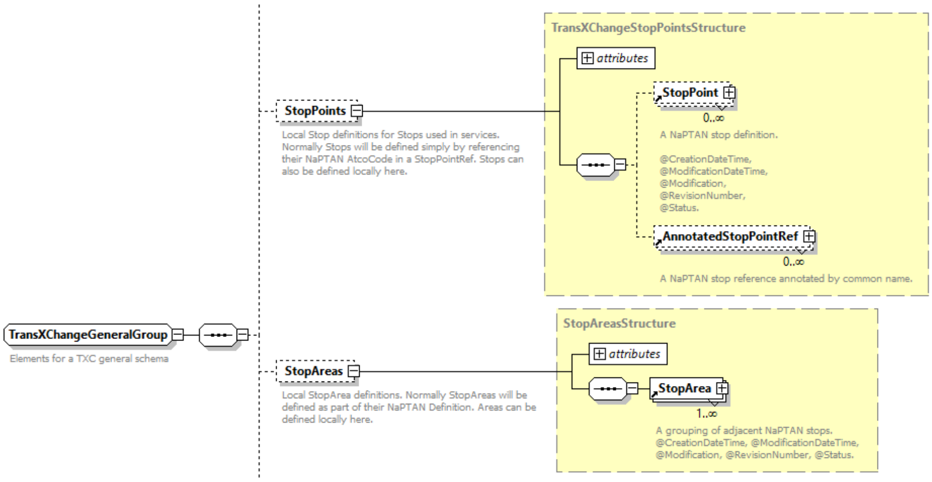

TransXChangeStopPointsStructure
attributes
StopPoint
0..∞
A NaPTAN stop definition.
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber,
@Status.
AnnotatedStopPointRef
0..∞
A NaPTAN stop reference annotated by common name.
StopPoints
Local Stop definitions for Stops used in services. Normally Stops will be defined simply by referencing their NaPTAN AtcoCode in a StopPointRef. Stops can also be defined locally here.
TransXChangeGeneralGroup
Elements for a TXC general schema
StopAreas
Local StopArea definitions. Normally StopAreas will be defined as part of their NaPTAN Definition. Areas can be defined locally here.
StopAreasStructure
attributes
StopArea
1..∞
A grouping of adjacent NaPTAN stops. @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber, @Status.

Figure 16 - StopPoints and StopAreas in a TXC document

Stop data (including for **StopAreas**) is maintained in NaPTAN as the primary source, and the TXC-PTI document shall always prefer to use a reference to a NaPTAN stop record; the creation of new stop records shall be reserved for the very specific cases set out below and shall be replaced by a NaPTAN reference as soon as one exists. Additionally, as stop areas are maintained in NaPTAN, **StopAreas** shall be omitted in TXC-PTI.

In TXC-PTI, **StopPoints** shall use **AnnotatedStopPointRef** elements to refer back to NaPTAN unless this is not possible. **StopPoint** shall be used sparingly and only for specific use-cases. **StopAreas** shall be omitted.

The use of StopPoint is allowed for a maximum period of 2 months. This is validated by ensuring the RouteLink referencing the StopPoint is not used by one or more VehicleJourney for more than 2 months by checking start and dates of the OperatingProfile for the VehicleJourney.

6.2 Stop points referencing NaPTAN

As already indicated, TXC-PTI shall as far as possible reference an existing NaPTAN stop, using an **AnnotatedStopPointRef**, which has the structure shown in the figure below.

The permitted elements are as follows:

Table 15 - Permitted usage of AnnotatedStopPointRef elements

Element Name	**Data Type**	**Used in TXC-PTI**
StopPointRef	*StopPointCodeType*	Mandated by schema
CommonName	*NaturalLanguageString*	Mandated by schema
Crossing	*NaturalLanguagePlaceName*	Not used – read from NaPTAN
Street	*NaturalLanguagePlaceName*	Not used – read from NaPTAN
Indicator	*NaturalLanguagePlaceName*	Where appropriate to aid readability
StopType	*StopTypeEnumeration*	Not used – read from NaPTAN
BusStopType	*BusStopTypeEnumeration*	Not used – read from NaPTAN
LocalityName	*NaturalLanguageString*	Where appropriate to aid readability
LocalityQualifier	*NaturalLanguageString*	Where appropriate to aid readability
Location	*LocationStructure*	Not used – read from NaPTAN

Of the elements that are included within the **AnnotatedStopPointRef**, only two are required by the schema: **StopPointRef**, which is the actual NaPTAN reference (Atcocode) of the NaPTAN stop; and **CommonName**, which is intended to be a re-statement of the name of the stop in NaPTAN in order to aid readability and diagnostics in a TXC file. Other optional elements, which were extended in TXC v2.4, contain more of the information from the NaPTAN file.

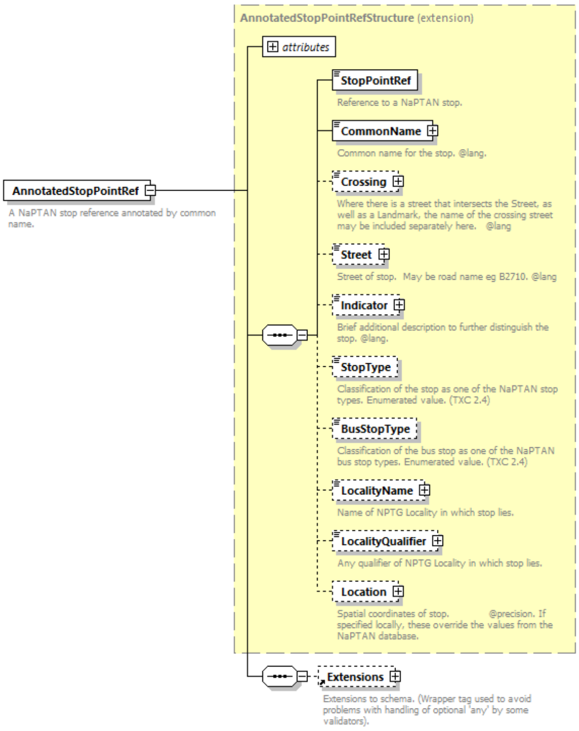

AnnotatedStopPointRefStructure (extension)
attributes
StopPointRef
Reference to a NaPTAN stop.
CommonName
Common name for the stop. @lang.
Crossing
Where there is a street that intersects the Street, as well as a Landmark, the name of the crossing street may be included separately here. @lang
Street
Street of stop. May be road name eg B2710. @lang
Indicator
Brief additional description to further distinguish the stop. @lang.
StopType
Classification of the stop as one of the NaPTAN stop types. Enumerated value. (TXC 2.4)
BusStopType
Classification of the bus stop as one of the NaPTAN bus stop types. Enumerated value. (TXC 2.4)
LocalityName
Name of NPTG Locality in which stop lies.
LocalityQualifier
Any qualifier of NPTG Locality in which stop lies.
Location
Spatial coordinates of stop. @precision. If specified locally, these override the values from the NaPTAN database.
AnnotatedStopPointRef
A NaPTAN stop reference annotated by common name.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 17 - The AnnotatedStopPointRef element

At no point was it intended that the elements populated in an **AnnotatedStopPointRef** structure should replace or overwrite the contents of NaPTAN.

This intent is continued, and made explicit, in TXC-PTI. **StopPointRef** is essential, and **CommonName** is mandated by the schema, so both shall be present in a TXC-PTI document.

Additionally, to aid readability, **Indicator**, **LocalityName** and **LocalityQualifier** may be included in a TXC-PTI document. Other elements shall be omitted and taken from NaPTAN.

In general, where a stop point field in TXC-PTI is different to NaPTAN, then the details held in NaPTAN shall be used in preference. If the naming in NaPTAN is deficient then the naming in NaPTAN should be discussed with the stop owner and agreement reached as to the accepted name.

Note that NaPTAN is a granular schema, with the different parts of the naming broken down into different fields. This allows users to compose stop names that suit their purposes from the different fields, which may be different depending on usage. For example, an express service between two towns, which only calls once in each locality, may well refer to the stop by just the locality name; conversely, a service which runs solely in one of those towns may well call the same stop by its common name only, and omit the locality completely. This is up to the end user, and data should not be overwritten simply to accommodate one use-case.

Any stop point that is used shall be appropriate to the service being described. For example, a bus service shall use only bus stops (either on-street or in bus stations) and shall not use rail station or ferry terminal stops, taxi ranks, etc. Likewise, where a TXC-PTI document is being used for other modes, those modes shall also use appropriate stops – metro service shall use metro stops, etc.

In TXC-PTI, an **AnnotatedStopPointRef** shall be used to refer to an appropriate NaPTAN stop. It shall contain a **StopPointRef** and a **CommonName**, and may optionally include an Indicator, a **LocalityName** and a **LocalityQualifier**. It shall not have other elements.

6.3 Stop points not referencing NaPTAN

As indicated earlier, there will be occasions where an operator needs to refer to a stop that is not yet in NaPTAN, or which has no real need to be in NaPTAN e.g. temporary stops for festivals, etc. In such cases, which will be set out in legislation and guidance, an operator may declare a new stop using a **StopPoint** element. This element embeds the NaPTAN schema, and so it shall be populated according to the rules set out in NaPTAN. The detail of this is outside of the scope of this document, and users should refer to the NaPTAN schema guidance document.

The use of StopPoint is checked by ensuring the RouteLink referencing the StopPoint is not used by one or more VehicleJourney for more than 2 months by checking start and dates of the OperatingProfile for the VehicleJourney

7 Routes and tracks

7.1 Introduction

In TXC, a **Route**, together with its constituent **RouteSections** and **RouteLinks**, describes the physical sequence of stopping points served. At its most basic it is a series of NaPTAN stop points and nothing more, which therefore provides a routing which is independent of any particular coordinate system. A particular route can be projected onto the underlying road network through the use of the **Tracks**. The following diagram from a route in Exeter serves to illustrate this:

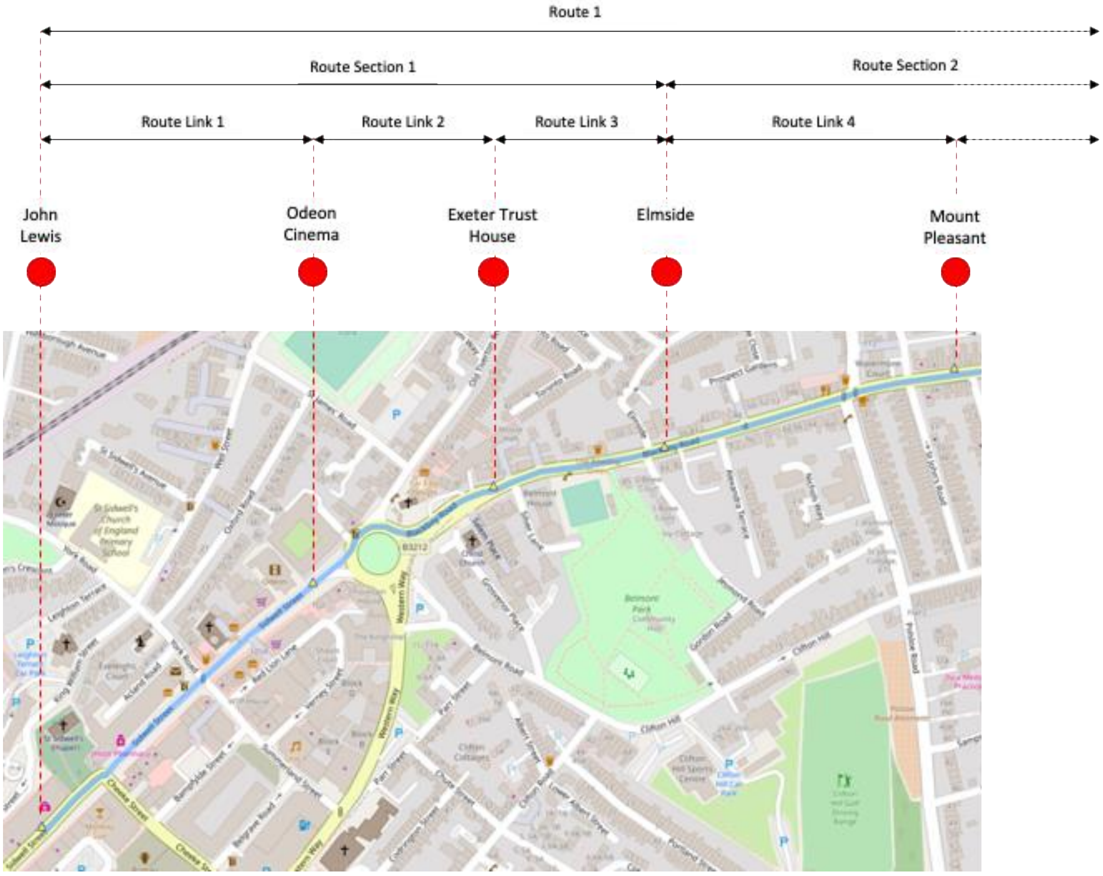


Figure 18 – Example of routes and tracks showing different levels

It is important to note that, within a TXC document, a route does not have any timing information associated with it. It is simply a series of stops which may be described in geographic terms by the addition of geographic points in a track. Because of this, and because a journey pattern restates the links that occur, a **Route** is an optional element in TXC, and several systems omit it.

Omitting track information, however, is limiting when the data will be used to drive real-time systems; prediction engines work best if they understand the route distance between stops as accurately as possible. Without track information, such systems will only be able to use “crow flies” distances between stops derived from stop coordinates in NaPTAN, or they will have to add information or estimate the tracks themselves. This is clearly undesirable, and hence **Route**, **RouteSection**, and **RouteLink** elements are mandated in TXC-PTI documents, with **Track** elements being strongly recommended.

These elements shall be populated in accordance with the TXC schema. Element ids can be constructed to suit the originating system, and there are no specific rules for TXC-PTI except that:

- The **RouteLink/Distance** element may be populated as an intermediate stage between no track and full track, although is shall be populated with an accurate measurement of the travelled distance between the two stops.
- The **RouteLink/Direction** element shall not be populated, in order to facilitate re-use of definitions between different routes without potential conflicting directions.

Additionally, a tendency in TXC is to export entire routes as single route sections, which then repeat sequences of stops when there are variations to routes e.g. short workings. It is strongly recommended that a TXC-PTI document uses efficiency when creating **RouteSections**, breaking them down into logical sections, so that re-use within **Routes** is readily achievable. This will help limit file sizes.

It is noted that operators submitting EBSR documents are required to provide tracks which have at least one point on every individual street that the vehicle travels down, in sequence. Where such information exists then this minimum information shall be used to populate a **Track**, although operators should strive to get full routing information into TXC-PTI as soon as practical.

The provision of Track information is important for providing high quality passenger information. It is used by real time information systems to calculate distances between stops to improve predicted times of arrival, the inclusion helps provide a high-quality presentation of a bus route on a map that follows the road rather than just straight line stop to stop.

In TXC-PTI, **Route**, **RouteSection**, and **RouteLink** elements shall be provided. It is strongly recommended that **RouteSections** are logically structured to facilitate re-use within **Routes** to help minimise file sizes.

In TXC-PTI, **Track** elements are strongly recommended. Where possible, operators should populate these with, at a minimum, the single point on each street as per the EBSR requirements.

The validator checks the RouteLink elements to ensure that no two RouteLinks are identical by checking AnnotatedStopPointRef, StopPoint and Track.

7.2 Routes

A **Route** element is mandated for TXC-PTI, as set out in the introduction to this section above. It's structure is as shown in Figure 19 below, and it shall use the elements as set out in .

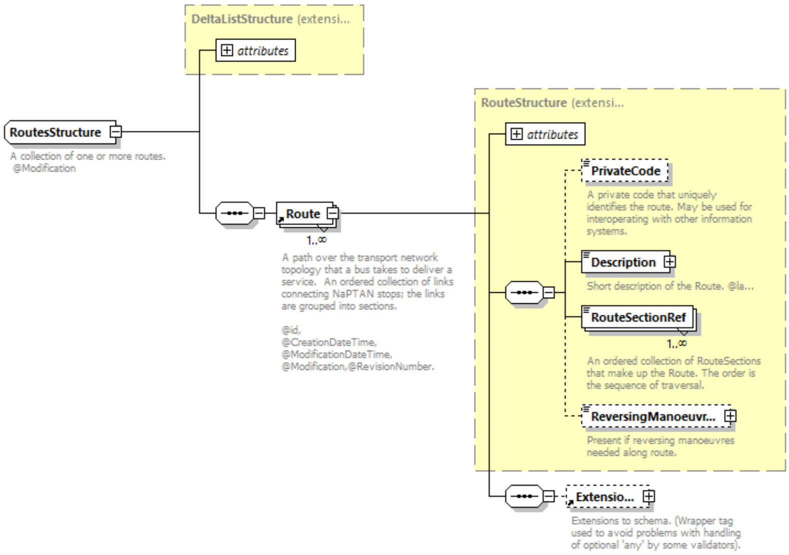

DeltaListStructure (extensi...
attributes
RoutesStructure
A collection of one or more routes. @Modification
Route
1..∞
A path over the transport network topology that a bus takes to deliver a service. An ordered collection of links connecting NaPTAN stops; the links are grouped into sections.
@id, @CreationDateTime, @ModificationDateTime, @Modification,@RevisionNumber.
RouteStructure (extensi...
attributes
PrivateCode
A private code that uniquely identifies the route. May be used for interoperating with other information systems.
Description
Short description of the Route. @la...
RouteSectionRef
1..∞
An ordered collection of RouteSections that make up the Route. The order is the sequence of traversal.
ReversingManoeuvr...
Present if reversing manoeuvres needed along route.
Extensio...
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 19 - The Routes structure

Table 16 - Usage of Route elements

Element Name	Data Type	Used in TXC-PTI
PrivateCode	*PrivateCodeType*	Where appropriate
Description	*NaturalLanguageString*	Mandated by schema
RouteSectionRef	*RouteSectionIfType*	References to an ordered list of **RouteSections**. Mandated by schema
ReversingManoeuvres	*NaturalLanguageString*	Not to be used

In TXC-PTI, a **Route** element shall be provided containing a **Description** and an ordered list of **RouteSectionRefs**. It may also contain a **PrivateCode** for compatibility with other systems if required. It shall not contain a **ReversingManoeuvres** element.

The validator checks to ensure that ReversingManoeuvres is not provided.

7.3 RouteSections and RouteLinks

At least one **RouteSection** element is mandated for TXC-PTI, as described in the introduction, and this in turn mandates at least one **RouteLink** within each **RouteSectionl**, although an ordered sequence of **RouteLink** elements is more usual. The structure of RouteLink is shown in the diagram below, with the permitted usages of elements shown in Table 17.

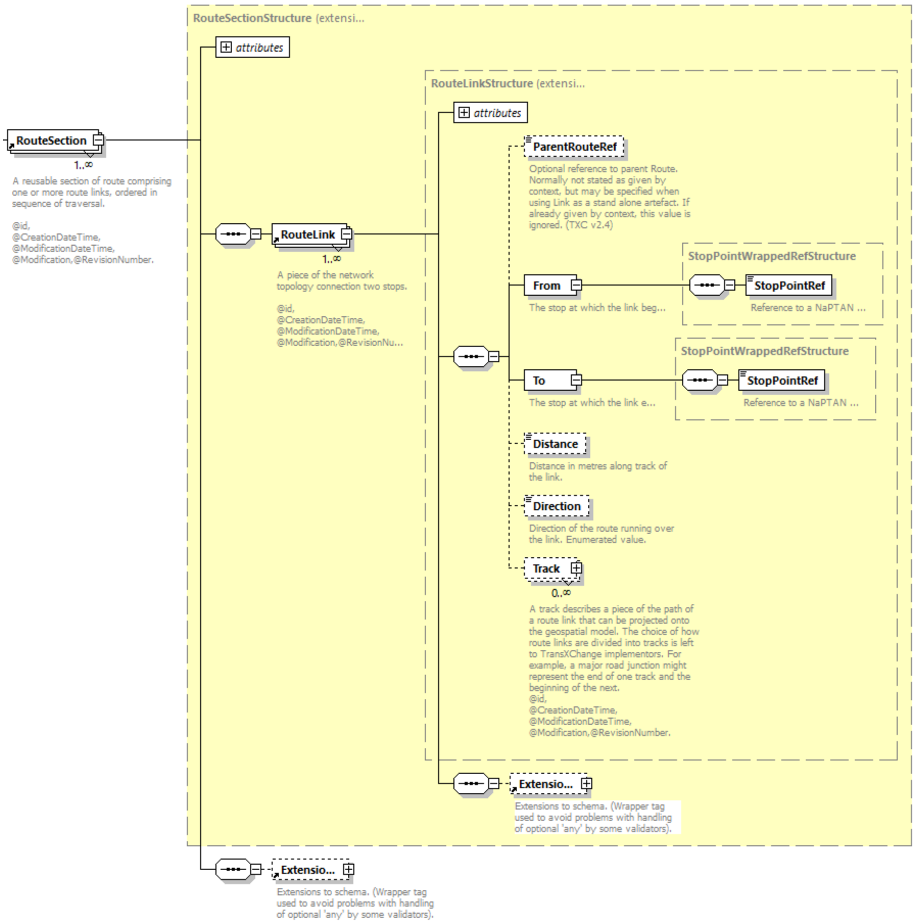

RouteSectionStructure (extensi...
attributes
RouteSection
1..∞
A reusable section of route comprising one or more route links, ordered in sequence of traversal.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,@RevisionNumber.
RouteLink
1..∞
A piece of the network topology connection two stops.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,@RevisionNu...
RouteLinkStructure (extensi...
attributes
ParentRouteRef
Optional reference to parent Route. Normally not stated as given by context, but may be specified when using Link as a stand alone artefact. If already given by context, this value is ignored. (TXC v2.4)
StopPointWrappedRefStructure
From
The stop at which the link beg...
StopPointRef
Reference to a NaPTAN ...
StopPointWrappedRefStructure
To
The stop at which the link e...
StopPointRef
Reference to a NaPTAN ...
Distance
Distance in metres along track of the link.
Direction
Direction of the route running over the link. Enumerated value.
Track
0..∞
A track describes a piece of the path of a route link that can be projected onto the geospatial model. The choice of how route links are divided into tracks is left to TransXChange implementors. For example, a major road junction might represent the end of one track and the beginning of the next.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,@RevisionNumber.
Extensio...
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
Extensio...
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 20 - RouteSection and RouteLink structure

Table 17 - Permitted usage of RouteLink elements

Element Name	Data Type	Used in TXC-PTI
ParentRouteRef	*RouteRefStructure*	Not used
From/StopPointRef	*StopPointRefStructure*	Mandated by schema
To/StopPointRef	*StopPointRefStructure*	Mandated by schema
Distance	*DistanceType*	Where appropriate
Direction	*LinkDirectionEnumeration*	Not to be used
Track	*TrackStructure*	Where appropriate. See 7.4

In TXC-PTI, **RouteSection**, and **RouteLinks** elements shall be provided. Each **RouteLink** shall contain “from” and “to” stop point references, and may contain a link **Distance** if appropriate. Link **Direction** is not to be used.

In TXC-PTI, **Track** elements are strongly recommended. Where possible, operators should populate these with, at a minimum, the single point on each street as per the EBSR requirements.

The validator checks to ensure that Direction is not provided.

7.4 Tracks

As already stated, there are very good reasons why a **Track** should be included in a TXC-PTI file. The track is an ordered set of coordinates that define the route that the vehicle takes between any given pair of stops. The structure of a **Track** is shown in the diagram below and, while it looks complicated, in practice it is not.

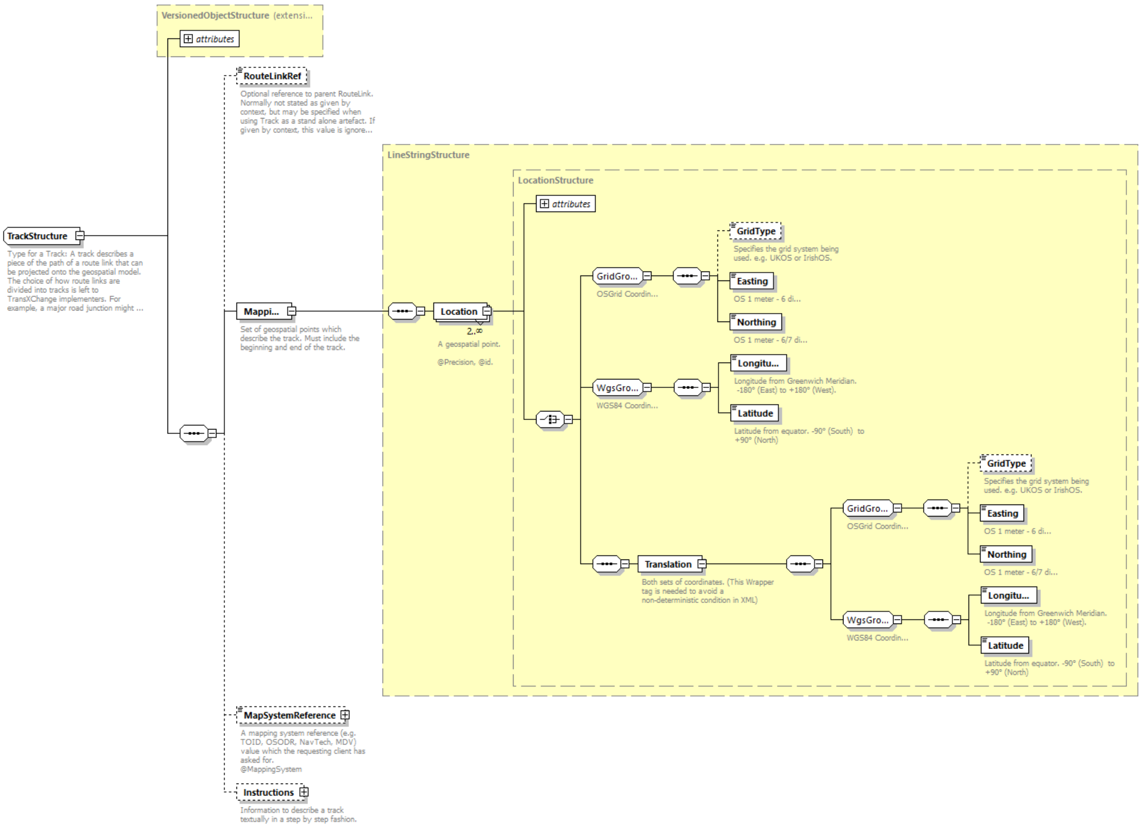

VersionedObjectStructure (extensi...
attributes
RouteLinkRef
Optional reference to parent RouteLink. Normally not stated as given by context, but may be specified when using Track as a stand alone artefact. If given by context, this value is ignore...
TrackStructure
Type for a Track: A track describes a piece of the path of a route link that can be projected onto the geospatial model. The choice of how route links are divided into tracks is left to TransXChange implementers. For example, a major road junction might ...
Mappi...
Set of geospatial points which describe the track. Must include the beginning and end of the track.
LineStringStructure
Location
2..∞
A geospatial point.
@Precision, @id.
LocationStructure
attributes
GridType
Specifies the grid system being used. e.g. UKOS or IrishOS.
GridGro...
OSGrid Coordin...
Easting
OS 1 meter - 6 di...
Northing
OS 1 meter - 6/7 di...
WgsGro...
WGS84 Coordin...
Longitu...
Longitude from Greenwich Meridian. -180° (East) to +180° (West).
Latitude
Latitude from equator. -90° (South) to +90° (North)
Translation
Both sets of coordinates. (This Wrapper tag is needed to avoid a non-deterministic condition in XML)
GridType
Specifies the grid system being used. e.g. UKOS or IrishOS.
GridGro...
OSGrid Coordin...
Easting
OS 1 meter - 6 di...
Northing
OS 1 meter - 6/7 di...
WgsGro...
WGS84 Coordin...
Longitu...
Longitude from Greenwich Meridian. -180° (East) to +180° (West).
Latitude
Latitude from equator. -90° (South) to +90° (North)
MapSystemReference
A mapping system reference (e.g. TOID, OSODR, NavTech, MDV) value which the requesting client has asked for.
@MappingSystem
Instructions
Information to describe a track textually in a step by step fashion.

Figure 21 - The Track structure

In essence, the diagram is showing that a **Track** can consist of *either* an X-Y grid reference (usually OSGR for the UK, although IrishOS is also permitted), *or* a lat-lon reference (using the WGS84 datum), *or* a Translation element which allows both types of reference to be specified at the same time. This latter is more useful to end users, as they can then work in their preferred coordinate system without requiring further translation of coordinates, and is therefore preferred in TXC-PTI. This may be beyond the scope of many users generating the data, however, so this is a strong recommendation rather than being mandatory.

Where a **Track** is provided, the usage of elements in TXC-PTI is as shown in the table below.

Table 18 - Permitted usage of Track elements

Element Name				**Data Type**	**Used in TXC-PTI**
RouteLinkRef				*RouteLinkRefStructure*	Not used – set by context
Mapping				*LineStringStructure*	Mandated by schema
	Location			*LocationStructure*	Mandated by schema. At least two **Location** elements required (one for each point on the track).
	(Grid)	GridType		*LocationGridTypeEnumeration*	Set to OSGR for clarity
		Easting		*EastingType*	Mandated by schema
		Northing		*NorthingType*	Mandated by schema
	(or WGS)	Longitude		*LongitudeType*	Mandated by schema
		Latitude		*LatitudeType*	Mandated by schema
	(or both)	Translation			Preferred option in TXC-PTI
		(all)	GridType	*LocationGridTypeEnumeration*	Set to OSGR for clarity
			Easting	*EastingType*	Mandated by schema
			Northing	*NorthingType*	Mandated by schema
			Longitude	*LongitudeType*	Mandated by schema
			Latitude	*LatitudeType*	Mandated by schema
MapSystemReference				*MapSystemReferenceStructure*	Optional
Instructions				*TrackInstructionsStructure*	Optional

In TXC-PTI, a single **Track** (if included) shall contain sufficient Location elements to define the route that the vehicle takes between a pair of stops.

In TXC-PTI, use of **Translation** to allow both OSGR and WGS84 coordinates to be defined is strongly recommended, but not mandatory. Users can provide coordinates in their preferred coordinate system only, if required.

Track data if provided is checked to ensure that it includes at least two Location elements.

8 Journey timings on standard services

8.1 Introduction

Journey timings in TXC are handled through **JourneyPatterns**, **JourneyPatternSections** and **JourneyPatternTimingLinks**. These relate structurally to the non-timed equivalents **Routes**, **RouteSections** and **RouteLinks**. TXC permits a **JourneyPattern** to operate over a shortened version of the **Route** that it maps onto, but for TXC-PTI this is not recommended and a **JourneyPattern** (and its sub-elements) shall have direct one-to-one mapping to the underlying **Route** (and its sub-elements).

As with routes, there has been a tendency in TXC to generate journey patterns which consist of a single journey pattern section with all of the timing links required for that particular journey pattern. Like routes, this leads to a significant increase in the file size since there is a high degree of redundancy. Therefore, TXC-PTI recommends that **JourneyPatterns**, like **Routes**, shall be structured efficiently so that sectional re-use is facilitated. This requirement in any case derives from the requirement to map journey patterns onto routes with a one-to-one mapping.

In TXC-PTI there shall be at least one **JourneyPattern** for each direction, that a service operates, of a non-flexible service (**StandardService –** see section 5.3.5), and this (and its sub-elements) shall have a one-to-one mapping to a **Route** (and its sub-elements).

In TXC-PTI it is strongly recommended that **JourneyPatternSections** are logically structured to facilitate re-use within **JourneyPatterns** to help minimise file sizes.

Note that TXC is a link-based schema. That means that a vehicle journey states the departure time of the trip, but that thereafter there is a duration over each successive link. This avoids problems where times can occur out of sequence in call-time based systems, but does mean that the actual call times at each stop need to be calculated by the consuming system by summing the link durations and any wait times up to that point and adding this to the departure time.

In TXC, there are two common ways of populating **JourneyPatternSections** (and the **JourneyPatternTimingLinks** that they contain):

1) There is a **JourneyPattern**, **JourneyPatternSection** and **JourneyPatternTimingLink** set for every different combination of link duration, wait time and stop activities, with each piece of information explicitly written in. Individual vehicle journeys reference the appropriate journey pattern; and

2) There is a single **JourneyPattern**, **JourneyPatternSection** and **JourneyPatternTimingLink** set for each direction, with each link duration and wait time set to zero minutes. This method uses the facilities in the vehicle journey to override the default zero-minute durations for each of the links. Stop activity, however, is usually included in the **JourneyPatternTimingLink** in the same way as method (1).

TXC-PTI will accept either way of defining the timings, so long as:

- only one method is used, not a mix of methods; and

- all run times and wait times in **JourneyPatternTimingLinks** in method 2 are set to zero minutes duration, and all vehicle journeys explicitly state the duration and wait time for each link.

For the avoidance of doubt, TXC-PTI permits the definition of the stop activity within the **JourneyPatternTimingLink**, as set out below, which shall be inherited by the vehicle journeys even when link durations are being overridden.

In TXC-PTI, either sufficient **JourneyPatternSection** elements shall exist to fully define the different timing and activities of a service, with all values explicitly stated, or there shall be a minimum of **JourneyPatternSection** elements where all of the durations and wait times are set to zero.

To check that only only one method of defining timings is being used if a JourneyPatternSection has non-zero times then the vehicle journey should not include link times. If this is not the case then the validator will raise an error.

8.2 The JourneyPattern element

A **JourneyPattern** element is constructed using a *JourneyPatternStructure*, as shown below.

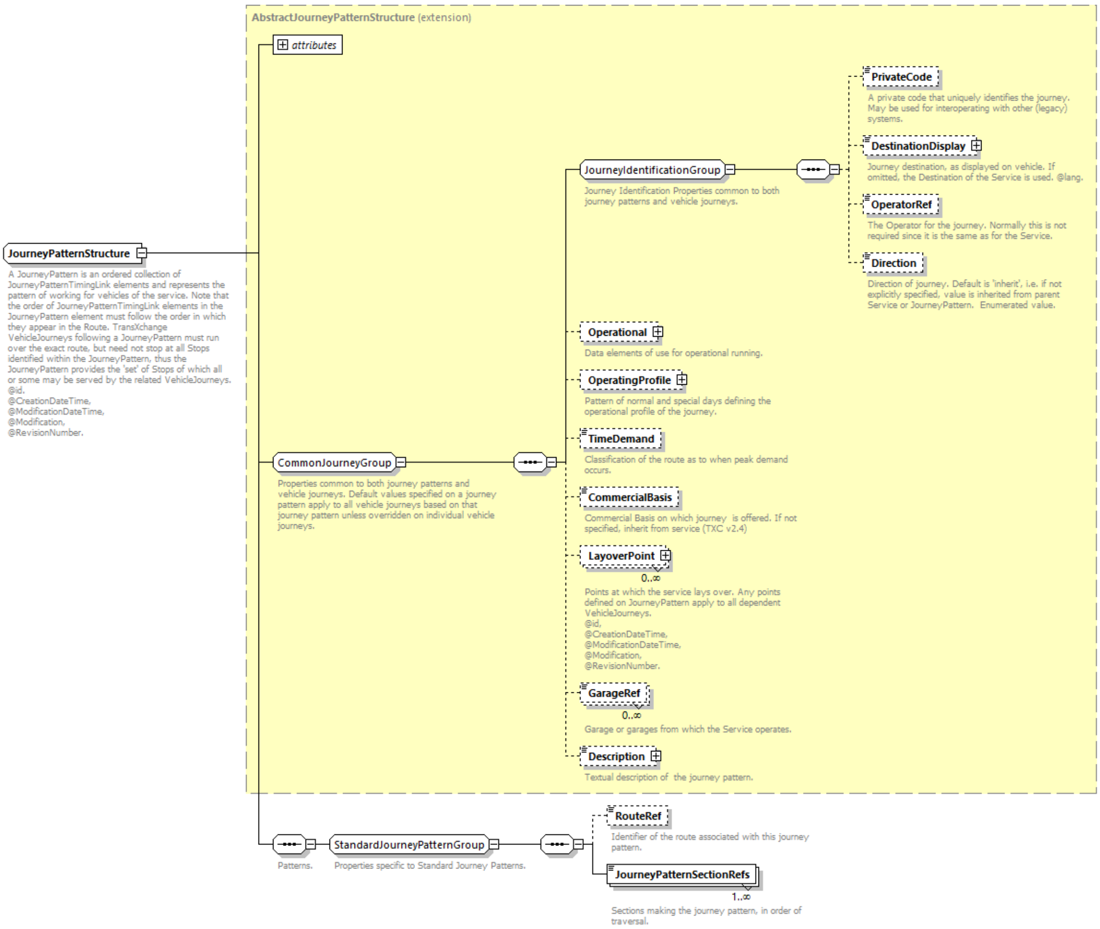

AbstractJourneyPatternStructure (extension)
attributes
JourneyPatternStructure
A JourneyPattern is an ordered collection of JourneyPatternTimingLink elements and represents the pattern of working for vehicles of the service. Note that the order of JourneyPatternTimingLink elements in the JourneyPattern element must follow the order in which they appear in the Route. TransXchange VehicleJourneys following a JourneyPattern must run over the exact route, but need not stop at all Stops identified within the JourneyPattern, thus the JourneyPattern provides the 'set' of Stops of which all or some may be served by the related VehicleJourneys.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
JourneyIdentificationGroup
Journey Identification Properties common to both journey patterns and vehicle journeys.
PrivateCode
A private code that uniquely identifies the journey. May be used for interoperating with other (legacy) systems.
DestinationDisplay
Journey destination, as displayed on vehicle. If omitted, the Destination of the Service is used. @lang.
OperatorRef
The Operator for the journey. Normally this is not required since it is the same as for the Service.
Direction
Direction of journey. Default is 'inherit', i.e. if not explicitly specified, value is inherited from parent Service or JourneyPattern. Enumerated value.
CommonJourneyGroup
Properties common to both journey patterns and vehicle journeys. Default values specified on a journey pattern apply to all vehicle journeys based on that journey pattern unless overridden on individual vehicle journeys.
Operational
Data elements of use for operational running.
OperatingProfile
Pattern of normal and special days defining the operational profile of the journey.
TimeDemand
Classification of the route as to when peak demand occurs.
CommercialBasis
Commercial Basis on which journey is offered. If not specified, inherit from service (TXC v2.4)
LayoverPoint
0..∞
Points at which the service lays over. Any points defined on JourneyPattern apply to all dependent VehicleJourneys.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
GarageRef
0..∞
Garage or garages from which the Service operates.
Description
Textual description of the journey pattern.
Patterns.
StandardJourneyPatternGroup
Properties specific to Standard Journey Patterns.
RouteRef
Identifier of the route associated with this journey pattern.
JourneyPatternSectionRefs
1..∞
Sections making the journey pattern, in order of traversal.

Figure 22 - The JourneyPattern structure

These elements shall be populated according to the table below.

Table 19 - Permitted elements for JourneyPattern

Element Name	**Data Type**	**Used in TXC-PTI**
PrivateCode	*PrivateCodeType*	Where appropriate
DestinationDisplay	*NaturalLanguageString*	Yes, shall be provided if not using **DynamicDestinationDisplay**.
OperatorRef	*OperatorRefStructure*	Yes, shall be included. Reference to defined **Operator**
Direction	*JourneyPatternVehicleDirection Enumeration*	Yes, shall be included.
Operational	*OperationalStructure*	Not used at this level. Use within **Service** instead. See section 5.3.3.
OperatingProfile	*OperatingProfileStructure*	Not used at this level. Use within **Service** or **VehicleJourney**. See sections 5.3.3 and 9.3.
TimeDemand	*TimeDemandEnumeration*	Not used at this level.
CommercialBasis	*JourneyCommercialBasisEnumeration*	Not used at this level. Use within **Service** or **VehicleJourney**. See sections 5.3.3 and 9.2.
LayoverPoint	*LayoverPointStructure*	May be included to aid in real time information. See section 9.7.2
GarageRef	*GarageRefStructure*	Not used at this level
Description	*NaturalLanguageString*	Optional
RouteRef	*RouteRefStructure*	Yes, shall be included. Reference to a defined **Route**
JourneyPatternSectionRefs	*JourneyPatternSectionRefStructure*	Yes, shall be included (at least one)

A **PrivateCode** may be included for compatibility with legacy systems.

A **DestinationDisplay** element shall be provided unless a DynamicDestinationDisplay is provided for all stops in the **JourneyPatternTimingLink**. This allows systems to state the ultimate destination of the journey.

An **OperatorRef** shall be included and shall reference the *id* of the (single) **Operator** element.

Direction shall be included at the **JourneyPattern** level. It shall then be assumed that all **RouteSections** and **RouteLinks** are in the same direction. This allows re-use without confusion of, for example, route sections describing a route through a town centre which is used in multiple directions.

Operational, **OperatingProfile** and **CommercialBasis** elements shall not be provided at the **JourneyPattern** level. These are best placed within individual vehicle journeys. An **OperatingProfile** default and a **CommercialBasis** default may additionally be provided at the **Service** level.

A **RouteRef** shall be provided to link the journey pattern to the corresponding route.

A series of **JourneyPatternSectionRefs** shall be provided, referencing the sequence of journey pattern sections that together make up the journey pattern. As stated elsewhere, the journey pattern and its associated elements should be structured so as to facilitate re-use of journey pattern sections. There shall therefore, ideally, be multiple **JourneyPatternSectionRefs** in any given **JourneyPattern**.

One or more **LayoverPoints** may be defined in the **JourneyPattern** to assist in real time information. **LayoverPoints** are referenced in dead runs in a **VehicleJourney** - see section 0. Where LayoverPoints are defined in a **JourneyPattern** then they shall not be redefined in a **VehicleJourney** that references that **JourneyPattern**.

Other elements shall be omitted as they contain no useful public-facing information.

In TXC-PTI, a **JourneyPattern** shall have an **OperatorRef**, **RouteRef**, **Destination** and an ordered sequence of **JourneyPatternSectionRefs**. It shall also have a **Destination**, if this is not otherwise handled through **DynamicDestinationDisplay**. Other elements shall not be included /as they either contain no useful public-facing information or are provided at other levels in the TXC-PTI document instead.

A TXC-PTI JourneyPattern may define one or more **LayoverPoints** to assist in real time information.

If neither JourneyPattern/DestinationDisplay nor JourneyPatternStopUsageStructure/DynamicDestinationDisplay is included then at this level Destination should be included otherwise a validation error will occur.

8.3 The JourneyPatternSection element

A **JourneyPatternSection** is made up of one or more **JourneyPatternTiming** link elements, in sequence. There are no additional constraints placed upon a **JourneyPatternSection** by TXC-PTI other than those already set out with regards to mapping to **RouteSection** elements and to structuring sections to facilitate re-use by routes.

The *id* of a **JourneyPatternSection** can be constructed to suit the originating system.

8.4 The JourneyPatternTimingLink element

8.4.1 Introduction

The **JourneyPatternTimingLink** element contains the details of the timing of the stop in the sequence, and also the details of the activity that is carried out at the stop: whether passengers are allowed to board and alight; whether or not there is a dwell time at the stop; etc.

8.4.2 Basic Structure of a JourneyPatternTimingLink

The basic structure of a **JourneyPatternTimingLink** is as shown in Figure 23 below. It has permitted elements in TXC-PTI as shown in the Table 20.

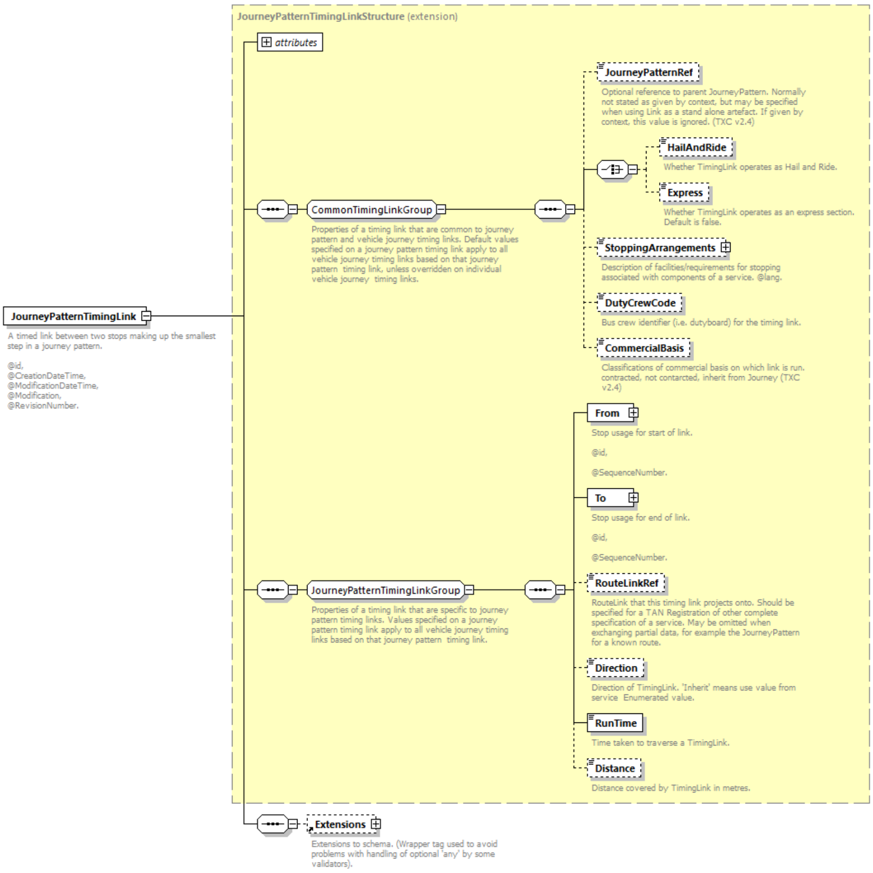

JourneyPatternTimingLinkStructure (extension)
attributes
JourneyPatternTimingLink
A timed link between two stops making up the smallest step in a journey pattern.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
CommonTimingLinkGroup
Properties of a timing link that are common to journey pattern and vehicle journey timing links. Default values specified on a journey pattern timing link apply to all vehicle journey timing links based on that journey pattern timing link, unless overridden on individual vehicle journey timing links.
JourneyPatternRef
Optional reference to parent JourneyPattern. Normally not stated as given by context, but may be specified when using Link as a stand alone artefact. If given by context, this value is ignored. (TXC v2.4)
HailAndRide
Whether TimingLink operates as Hail and Ride.
Express
Whether TimingLink operates as an express section. Default is false.
StoppingArrangements
Description of facilities/requirements for stopping associated with components of a service. @lang.
DutyCrewCode
Bus crew identifier (i.e. dutyboard) for the timing link.
CommercialBasis
Classifications of commercial basis on which link is run. contracted, not contarcted, inherit from Journey (TXC v2.4)
JourneyPatternTimingLinkGroup
Properties of a timing link that are specific to journey pattern timing links. Values specified on a journey pattern timing link apply to all vehicle journey timing links based on that journey pattern timing link.
From
Stop usage for start of link.
@id,
@SequenceNumber.
To
Stop usage for end of link.
@id,
@SequenceNumber.
RouteLinkRef
RouteLink that this timing link projects onto. Should be specified for a TAN Registration of other complete specification of a service. May be omitted when exchanging partial data, for example the JourneyPattern for a known route.
Direction
Direction of TimingLink. 'Inherit' means use value from service Enumerated value.
RunTime
Time taken to traverse a TimingLink.
Distance
Distance covered by TimingLink in metres.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 23 - Overview of JourneyPatternTimingLink element

Table 20 - Permitted elements in JourneyPatternTimingLink

Element Name	Data Type	Used in TXC-PTI
JourneyPatternRef	*JourneyPatternRefStructure*	Not used
HailAndRide	*boolean*	Not used
Express	*boolean*	Not used
StoppingArrangements	*NaturalLanguageString*	Not used
DutyCrewCode	*DutyCrewCodeType*	Omitted at this level. Use in **VehicleJourney**. See 0
CommercialBasis	*LinkCommercialBasisEnumeration*	Not used at this level. Use within **Service** or **VehicleJourney**. See sections 5.3.3 and 9.3
From	*JourneyPatternStopUsageStructure*	Mandated by schema. See 8.4.3
To	*JourneyPatternStopUsageStructure*	Mandated by schema. See 8.4.3
RouteLinkRef	*RouteLinkRefStructure*	Yes, shall be included.

Direction	*JourneyPatternLinkDirection Enumeration*	Not used at this level.
RunTime	*DurationType*	Mandated by schema. Zero or value. See section 8.1
Distance	*DistanceType*	Where appropriate

For TXC-PTI, a **JourneyPatternTimingLink** shall be populated as follows:

- **RouteLinkRef** shall be populated with the id of the corresponding **RouteLink**.
- **RunTime** shall be populated either with the actual value of the run time over the link, or with a duration of zero minutes, as described in section 8.1.
- **Distance** may be populated if available. If it is populated, then it shall have the same value as the any **Distance** element or length of **Track** on the corresponding **RouteLink**.
- **DutyCrewCode** shall be omitted, as this is better placed in individual vehicle journeys.
- **Direction** shall be omitted from a **JourneyPatternTimingLink**. This permits reuse of sections and links where, for example, a section of a service is used in two different directions such as a loop through a town centre due to one-way systems.
- **JourneyPatternRef**, **HailAndRide**, **Express**, **StoppingArrangements** and **CommercialBasis** shall be omitted. The reference to the journey pattern is implied, and the other elements do not contain useful public-facing information. There might be some merit in describing a link as **HailAndRide**, but this would require detailed track information which is only recommended rather than mandatory, and it would have to apply to the whole of the link even if only part of it was hail and ride. Instead, specify hail and ride sections by reference to NaPTAN “virtual” hail and ride stops which have starts and ends of hail and ride sections defined.

In TXC-PTI, a **JourneyPatternTimingLink** shall have a **RouteLinkRef** and **RunTime**, and may have a **Distance**. It shall additionally have **From** and **To** elements but shall not have other elements. Specifically, it shall not have a **Direction** element.

The validator checks to ensure that Direction is not provided.

8.4.3 Stop definition and wait times

Every **JourneyPatternTimingPointLink** requires a **From** element, detailing the departure stop and activity, and a **To** element, detailing the arrival stop and activity. These two elements are constructed the same way, using a *JourneyPatternStopUsageStructure*.

In versions of TXC prior to v2.4, the intent of these structures was that they were additive. That is, they combined to give an overall effect for activity at the stop, remembering that the arrival stop (**To** element) on one link is the same physical stop point as the departure stop (**From** element) on the next successive link. So, for example, if a **To** element defined a stop as “set down only” and a **From** element defined a stop as “pick up only” then the overall effect was that it allowed both setting down and picking up.

One area where this caused particular confusion was with wait times. Both arrival and departure settings could define a wait time – the intention was, originally, to be able to specify an unloading time and a loading time separately, although in practice this rarely if ever happened.

In TXC v2.4, this behaviour fundamentally changed. The schema guidance, unfortunately, did not reflect this, but it is clear from the schema. From version 2.4 onwards, the wait time was to be the total wait time and was to be stated identically on both the **To** element of the incoming link and the **From** element of the next, outgoing link.

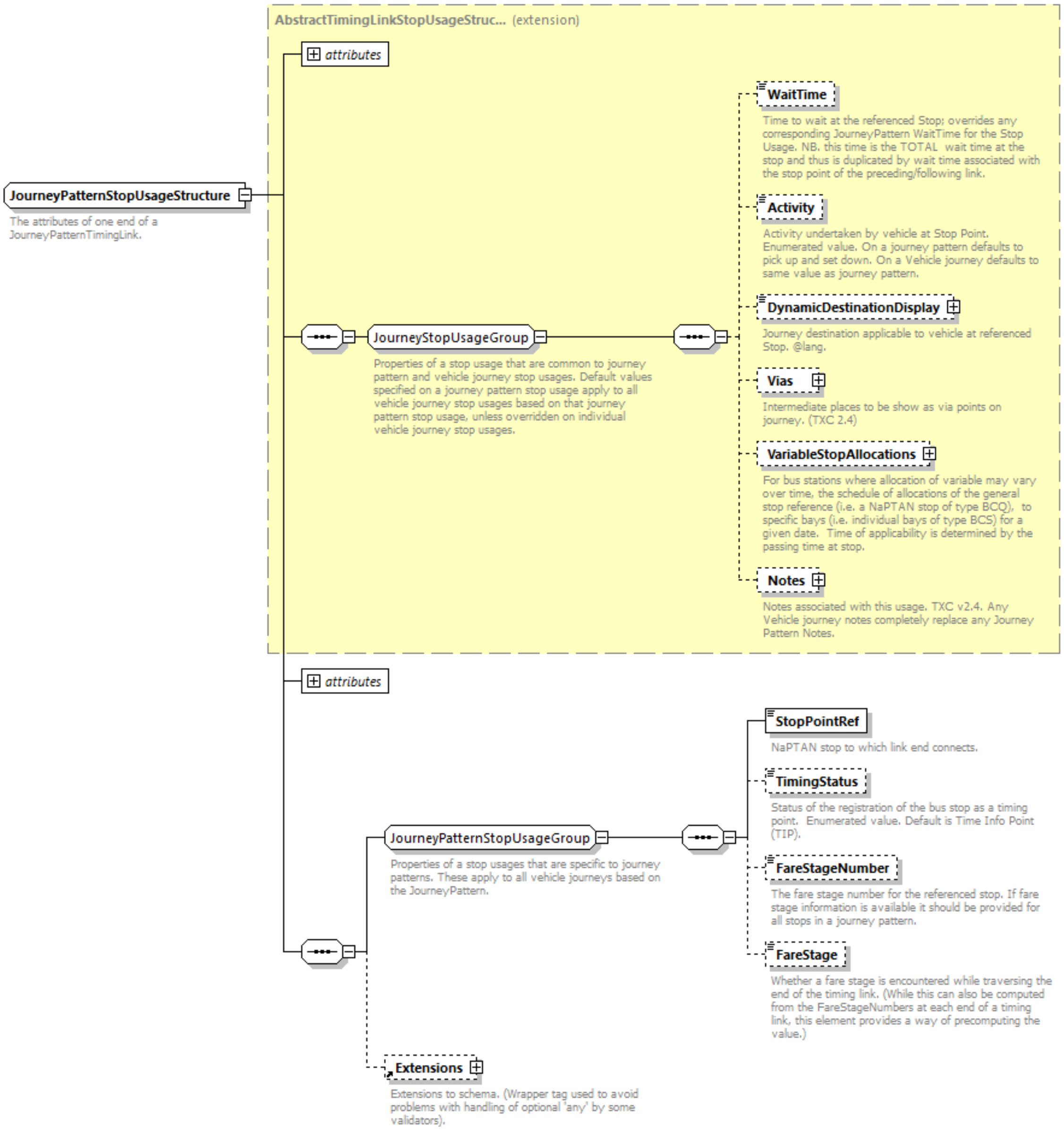

AbstractTimingLinkStopUsageStruc... (extension)
attributes
JourneyPatternStopUsageStructure
The attributes of one end of a JourneyPatternTimingLink.
JourneyStopUsageGroup
Properties of a stop usage that are common to journey pattern and vehicle journey stop usages. Default values specified on a journey pattern stop usage apply to all vehicle journey stop usages based on that journey pattern stop usage, unless overridden on individual vehicle journey stop usages.
WaitTime
Time to wait at the referenced Stop; overrides any corresponding JourneyPattern WaitTime for the Stop Usage. NB. this time is the TOTAL wait time at the stop and thus is duplicated by wait time associated with the stop point of the preceding/following link.
Activity
Activity undertaken by vehicle at Stop Point. Enumerated value. On a journey pattern defaults to pick up and set down. On a Vehicle journey defaults to same value as journey pattern.
DynamicDestinationDisplay
Journey destination applicable to vehicle at referenced Stop. @lang.
Vias
Intermediate places to be show as via points on journey. (TXC 2.4)
VariableStopAllocations
For bus stations where allocation of variable may vary over time, the schedule of allocations of the general stop reference (i.e. a NaPTAN stop of type BCQ), to specific bays (i.e. individual bays of type BCS) for a given date. Time of applicability is determined by the passing time at stop.
Notes
Notes associated with this usage. TXC v2.4. Any Vehicle journey notes completely replace any Journey Pattern Notes.
attributes
JourneyPatternStopUsageGroup
Properties of a stop usages that are specific to journey patterns. These apply to all vehicle journeys based on the JourneyPattern.
StopPointRef
NaPTAN stop to which link end connects.
TimingStatus
Status of the registration of the bus stop as a timing point. Enumerated value. Default is Time Info Point (TIP).
FareStageNumber
The fare stage number for the referenced stop. If fare stage information is available it should be provided for all stops in a journey pattern.
FareStage
Whether a fare stage is encountered while traversing the end of the timing link. (While this can also be computed from the FareStageNumbers at each end of a timing link, this element provides a way of precomputing the value.)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 24 - The JourneyPatternStopUsage structure

TXC-PTI is making this requirement explicit. Moreover, there are additional benefits to making as many of the other elements as possible behave the same way, as this aids readability and is unequivocal as to what is happening at the stop. Some elements still cannot be treated this way, however, and these are outlined below.

A **From** or **To** element in a **JourneyPatternTimingLink** in TXC-PTI shall therefore use the elements and values for a **JourneyPatternStopUsageStructure** shown in the table below.

Table 21 - Permitted elements and values in a TXC-PTI JourneyPatternStopUsageStructure

Element Name	**Data Type**	**Use in TXC-PTI**
WaitTime	*DurationType*	The total wait time at the stop, which is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link. Shall be included where appropriate.
Activity	*VehicleAtStopActivity Enumeration*	The activity at the stop, which is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link. Shall always be included if the behaviour is not the default pickUpAndSetDown. Permitted values are *pickup*, *setDown*, *pickUpAndSetDown* and *pass*. The majority of stops will inherit the TXC default of *pickUpAndSetDown.*
DynamicDestinationDisplay	*NaturalLanguageString*	A textual description of the destination of the vehicle which may be shown at this stop. This is different from the overall destination of the journey pattern and can vary along the route. It is particularly useful for circular services where, for example, the destination shown at the first stop would otherwise be the same stop. Shall always be included if neither **JourneyPattern/DestinationDisplay** nor VehicleJourney/Destination is included and shall be included if appropriate otherwise.
Vias	*ViasStructure*	Not used.
VariableStopAllocations	*VariableStopAllocations Structure*	Not used.
Notes	*NotesStructure*	May be included where appropriate. See section 2.5. May be different on the **To** end of the incoming link and the **From** end of the next, outgoing link.
StopPointRef	*StopPointRefStructure*	The AtcoCode of a referenced NaPTAN stop point defined in an **AnnotatedStopPointRef**, or the pseudo-AtcoCode of the operator-defined stop point defined in a **StopPoint**. It is duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link as it is the same stop.

TimingStatus	*TimingStatusEnumeration*	Defines the reliability of the timing at the stop and may also be used as a flag to aid in the display of "main point" timetables. Shall always be included. TXC v2.4 includes both the older three-letter codes as well as newer, more readable versions. To aid readability, only those values set out below shall be used in a TXC-PTI document. Allowed values are: *principalTimingPoint* (note: only this spelling) *timeInfoPoint* *otherPoint*
FareStageNumber	*FareStageNumberType*	States the fare stage number for this stop. May be different on the **To** end of the incoming link and the **From** end of the next, outgoing link. May be included where appropriate.
FareStage	*boolean*	A true/false flag which provides a simple way to determine if a fare stage is changing at this stop. It is to be duplicated on the **To** end of the incoming link and the **From** end of the next, outgoing link. May be included where appropriate.

In TXC-PTI, a **From** element and the **To** element on the next link shall have the same information for the stop, except for those elements identified above.

Within JourneyPatternSections, if there are two JourneyPatternTiminglinks, A and B that follow on from each other. The details about the 'To' stop from link A shall be the same as for the 'From' stop in link B.

If neither JourneyPattern/DestinationDisplay nor VehicleJourney/Destination is included, then DynamicDestinationDisplay at JourneyPatternStopUSageStructure level should be included otherwise a validation error will occur.

Tests to ensure that the Activity for a stop is set correctly are included in the Data Quality tests. This identifies where the first stop is designated as set down only and the last stop is designated to be pick up only which are incorrect.

8.4.4 Formatting for timetable displays

One of the common problems with TXC data is how disparate trips on different days of the week can be structured into a traditional matrix timetable, and one of the goals of the TXC-PTI profile was to facilitate this where possible.

To achieve this, a TXC-PTI document shall use the *sequenceNumber* attribute of a **JourneyPatternTimingLink**.

The sequence number is not, as has sometimes been seen, simply a number which represents the order of links in a particular **JourneyPatternSection**. Rather, it is a number which represents that stop's position in a row on a timetable. This is best illustrated by considering the following timetable:

Table 22 - Example of a structured timetable

Seq. No.	Name	AtcoCode	Trip 1	Trip 2
1	One	1580ABCD	08:00	08:15
2	Two	1580EFGH	08:02	08:17
3	Three	1580JKLM	|	08:23
4	Four	1580NPQR	08:07	-

In TXC-PTI, this shall be coded in this way:

```
<JourneyPatternSection id="ABC1_jps1">
   <JourneyPatternTimingLink id="ABC1_jps1_tl1">
      <From SequenceNumber="1">
         <StopPointRef>1580ABCD</StopPointRef>
      </From>
      <To SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </To>
      <RunTime>PT2M</RunTime>
   </JourneyPatternTimingLink>
   <JourneyPatternTimingLink id="ABC1_jps1_tl2">
      <From SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </From>
      <To SequenceNumber="4">
         <StopPointRef>1580NPQR</StopPointRef>
      </To>
      <RunTime>PT5M</RunTime>
   </JourneyPatternTimingLink>
</JourneyPatternSection>
<JourneyPatternSection id="ABC1_jps2">
   <JourneyPatternTimingLink id="ABC1_jps2_tl1">
      <From SequenceNumber="1">
         <StopPointRef>1580ABCD</StopPointRef>
      </From>
      <To SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </To>
      <RunTime>PT2M</RunTime>
   </JourneyPatternTimingLink>
   <JourneyPatternTimingLink id="ABC1_jps2_tl2">
      <From SequenceNumber="2">
         <StopPointRef>1580EFGH</StopPointRef>
      </From>
      <To SequenceNumber="3">
         <StopPointRef>1580JKLM</StopPointRef>
      </To>
```

```
           <RunTime>PT6M</RunTime>
        </JourneyPatternTimingLink>
     </JourneyPatternSection>
```

Note the *SequenceNumber* usage in the **From** and **To** elements; it is clear from the numbering used which order the stops should appear, and which times shall appear against which stops.

In TXC-PTI, a **JourneyPatternTimingLink** shall use the *SequenceNumber* attribute in the **From** and **To** elements in order to facilitate the correct layout of a timetable.

The validator tests checks that within the JourneyPatternTimingLink, the "From SequenceNumber" and "To SequeneceNumber" are populated.

9 Vehicle journeys

9.1 Introduction

In TXC, the **VehicleJourney** element defines the actual trip that a vehicle will make. Up to this point, the TXC has been defining the basic parameters of the service, the routings of the vehicles, the default timings of the links, etc., but this is the section of the TXC document where the actual departures are specified. It is essential that this information is correctly structured and reflects what the vehicle will do.

TXC-PTI adopts the approach that the best place to hold the information about a trip is within the trip itself. To that end it recommends that much of the detail about the trip, and in particular the dates on which the trip operates, be contained within the **VehicleJourney** and not elsewhere. However, for reasons of readability and compactness it is possible to define default patterns and timings that suit the majority of the trips, and then override them in individual vehicle journeys where appropriate. In this case, though, complete sets of information shall be overridden, and the systems shall not rely on inheriting only parts of data elements as this leads to confusion and errors of interpretation.

9.2 Basic structure of a vehicle journey

9.2.1 Introduction

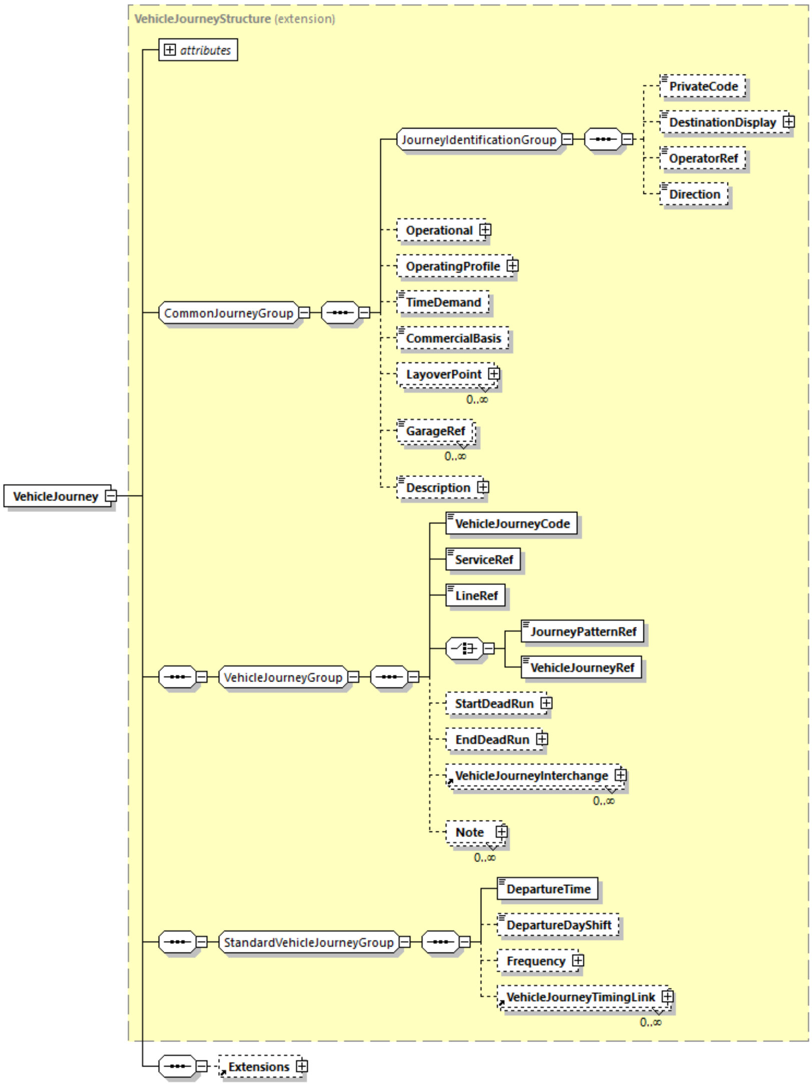


Figure 25 - Basic structure of a VehicleJourney

A vehicle journey record is structured as shown in the figure above. Many of these elements are described in more detail in the sections that follow. In general terms, however, a **VehicleJourney** in TXC-PTI has the usage outlined in the table below.

Table 23 - Usage of elements in a VehicleJourney

Element Name	Data Type	Used in TXC-PTI
PrivateCode	*PrivateCodeType*	Where appropriate
DestinationDisplay	*NaturalLanguageString*	Yes, shall be provided if not using **DestinationDisplay** or **DynamicDestinationDisplay** at a **JourneyPattern** level.
OperatorRef	*OperatorRefStructure*	Yes, shall be included. Reference to defined **Operator**
Direction	*JourneyPatternVehicleDirection Enumeration*	Not used at this level.
Operational	*OperationalStructure*	Where appropriate, shall be included to support real time information. See section 9.7. May also includes **VehicleType** information – see section 2.4.2
OperatingProfile	*OperatingProfileStructure*	Where appropriate to replace any definition in **Service**. See section 9.3
TimeDemand	*TimeDemandEnumeration*	Optional
CommercialBasis	*JourneyCommercialBasis Enumeration*	Where appropriate
LayoverPoint	*LayoverPointStructure*	Where appropriate, shall be included to support real time information. See section 9.7.2
GarageRef	*GarageRefStructure*	Where appropriate, shall be included to support real time information. See section 9.7
Description	*NaturalLanguageString*	Optional
VehicleJourneyCode	*VehicleJourneyCodeType*	Mandated by schema
ServiceRef	*ServiceRefStructure*	Mandated by schema
LineRef	*LineIdType*	Mandated by schema
(one of) JourneyPatternRef	*JourneyPatternIdType*	Mandated by schema (if VehicleJourneyRef not present)
VehicleJourneyRef	*VehicleJourneyCodeType*	Mandated by schema (if JourneyPatternRef not present)
StartDeadRun	*DeadRunStructure*	Where appropriate, shall be included to support real time information. See section 9.7.3
EndDeadRun	*DeadRunStructure*	Where appropriate, shall be included to support real time information. See section 9.7.3
VehicleJourneyInterchange	*VehicleJourneyInterchange Structure*	Where appropriate. See section 9.6
Note	*NoteStructure*	Where appropriate.
DepartureTime	*time*	Mandated by schema

DepartureDayShift	*integer*	Where appropriate. If present, shall be +1. See section 9.5
Frequency	*FrequentServiceStructure*	Where appropriate. No special rules in TXC-PTI; refer to TXC schema guidance.
VehicleJourneyTimingLink	*VehicleJourneyTimingLink Structure*	Where appropriate for providing a complete set of timing links at VehicleJourney level. See section 9.4

A TXC-PTI **VehicleJourney**:

- shall have a **VehicleJourneyCode**, a **ServiceRef** and a **LineRef** as mandated by the schema. TXC-PTI does not specify how a **VehicleJourneyCode** is to be structured, and this can be done to suit the originating system. It may however be useful to include the line ref as part of the code.
- shall have a **DestinationDisplay** where neither this nor a **DynamicDestinationDisplay** is provided at a **JourneyPattern** level.
- shall have either a **JourneyPatternRef**, referencing a **JourneyPattern**, or a **VehicleJourneyRef**, referencing another **VehicleJourney**.

  In general, most VehicleJourneys would refer to a JourneyPattern via a JourneyPatternRef. This allows the VehicleJourney to refrence any timings declared in that JourneyPattern and to either use the default OperatingProfile or provide a complete replacement for it.

  However, in some cases and particularly where another VehicleJourney has already been defined with the correct days of operation and a full set of times, the VehicleJourney can reference that other vehicle journey's VehicleJourneyCode using a VehicleJourneyRef.

  If a **VehicleJourneyRef** is used then all the details of the referenced **VehicleJourney**, including operating profile, timings, vehicle types, etc. shall be inherited and not overridden. The only elements that may be amended are the operational elements for real time systems.
- shall have a **DepartureTime** to say what time the **VehicleJourney** starts, and may optionally shift trips to the next day using **DepartureDayShift** (see section 0) or define them as frequent journeys using **Frequency** (refer to TXC schema guidance for specifying frequent journeys).
- may override the operating days by providing a complete set of days and dates when the trip operates using an **OperatingProfile** block (section 0)
- may override the vehicle timings by providing a complete set of times over all of the links in a journey pattern in a set of **VehicleJourneyTimingLink** elements (see section 9.4).
- shall have an **Operational** block and set of **GarageRef** elements containing information to support real time systems, where such information is available (see section 0).
- may include dead runs (as **StartDeadRun** and **EndDeadRun**) where appropriate (see section 0).
- may have a **PrivateCode** for compatibility with legacy systems.
- may include information on interchanges where appropriate (see section 9.6).

- may include a **CommercialBasis** flag to denote whether or not the service is operated commercially (notContracted) or whether it is wholly or partially contracted (contracted, partContracted). While this is of limited use to the public, contracting authorities are keen to demonstrate their role in providing socially necessary services. This flag is optional, and may have a default set at **Service** level. At this level, it denotes that this specific trip (and any others referencing it via **VehicleJourneyRef**) has the stated commercial basis.
- may define one or more **LayoverPoints** to assist in real time information. **LayoverPoints** are referenced in dead runs (either **StartDeadRun** or **EndDeadRun**). See section 9.7.2. **LayoverPoints** shall only be defined in a **VehicleJourney** if not present in the referenced **JourneyPattern** or referenced **VehicleJourney**.

In TXC-PTI, a **VehicleJourney** shall define the operation of a particular trip, optionally overriding some of the blocks so long as these are complete replacements. Some elements are not to be used.

If a VehicleJourneyRef is provided within a ‘VehicleJourney’, then ‘OperatingProfile’ is not allowed and will raise an error if provided.

If VehicleJourneyTimingLink is provided in VehicleJourney,then the number of links provided should be same as number of JourneyPatternTimingLinks provided in JourneyPattern otherwise an error is raised.

If DestinationDisplay or DynamicDestinationDisplay at a JourneyPattern level is not used, then DestinationDisplay is a mandatory element within VehicleJourney and if not provided will raise an error in the validator.

VehicleJourneyCode shall be used in the bus location data feed as VehicleJourneyRef to enable production of real time customer information.

9.2.2 Formatting for timetable displays

As with stops read down a timetable, one of the common problems with TXC data is how disparate trips on different days of the week can be structured across a traditional matrix timetable, and one of the goals of the TXC-PTI profile was to facilitate this where possible.

To achieve this, a TXC-PTI document may use the *sequenceNumber* attribute of a **VehicleJourney**.

The sequence number is not, as has sometimes been seen, simply a number which represents the order of the records in a particular set of **VehicleJourneys**. Rather, it is a number which represents that trip's position in a column on a timetable. This is best illustrated by considering the same timetable as before:

Table 24 - Example of a structured timetable

Seq. No.	Name	AtcoCode	Trip 1	Trip 2
1	One	1580ABCD	08:00	08:15
2	Two	1580EFGH	08:02	08:17
3	Three	1580JKLM	|	08:23
4	Four	1580NPQR	08:07	-

In TXC-PTI, this shall be coded in this way:

```
<VehicleJourneys>
   <VehicleJourney SequenceNumber="1" >
      <VehicleJourneyCode>Trip_1</VehicleJourneyCode>
      ...
   </VehicleJourney>
   <VehicleJourney SequenceNumber="2">
      <VehicleJourneyCode>Trip_2</VehicleJourneyCode>
      ...
   </VehicleJourney>
   ...
</VehicleJourneys>
```

Note the *SequenceNumber* usage in the **VehicleJourney** elements; it is clear from the numbering used which order the trips should appear. However, care does have to be taken when creating sequence numbers to give some thought as to how the trips are to be presented, because the order can vary depending on how trips are to be combined – for example, on a “Mondays to Saturdays” timetable compared to separate “Mondays to Fridays” and “Saturdays” timetables. For this reason, sequence numbers on vehicle journeys are optional in TXC-PTI, although they are recommended.

In TXC-PTI, a **VehicleJourney** may use the *SequenceNumber* in order to facilitate the correct layout of a timetable. This is recommended.

9.3 The OperatingProfile

9.3.1 Basic structure of an OperatingProfile

An OperatingProfile details the regular and irregular operation of a trip.

It consists of a *NormalDayPropertiesGroup*, which defines the regular operation, broken down into regular days which are then qualified (restricted) by periodic elements (e.g. 1st Wednesday of each month) or by serviced organisation dates (such as school term dates).

In addition to the regular days, elements also exist to add or remove specific date ranges, and to add or remove specific bank holidays (either enumerated or user-defined). Each of these has days of operation, which are in addition to any defined regular days, and days of non-operation, which remove any previously defined regular days.

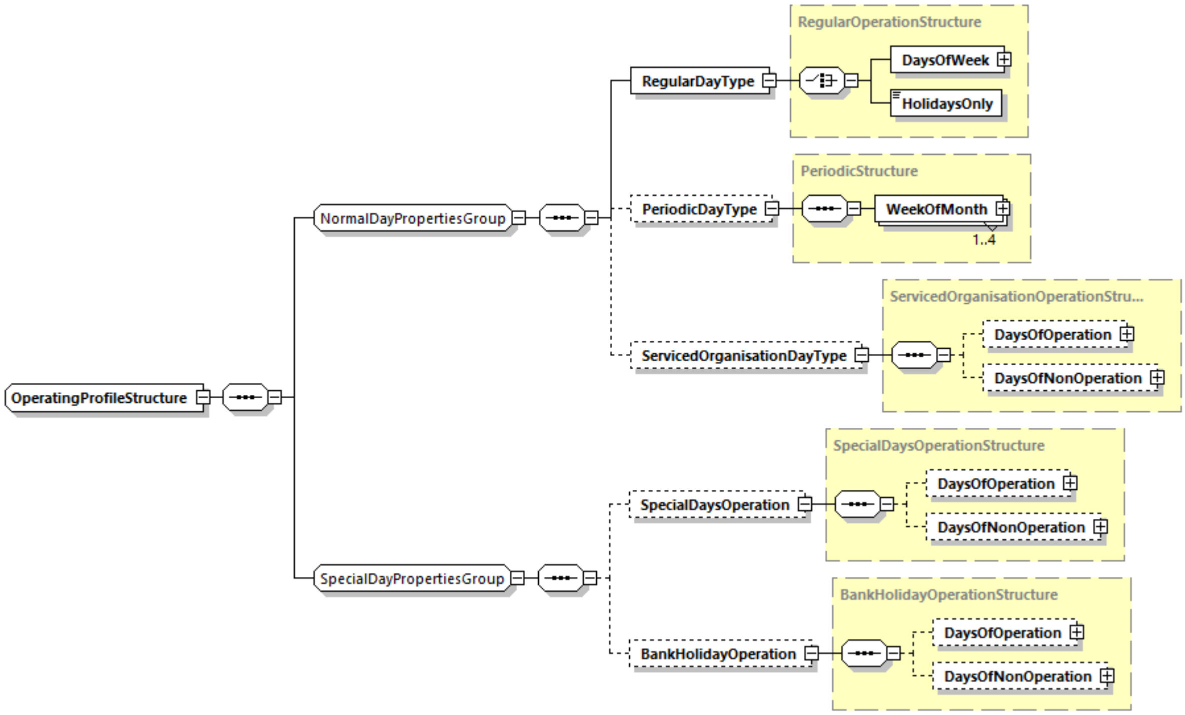


Figure 26 - Basic structure of an OperatingProfile

As with TXC, in the event of conflict between days of operation and days of non-operation, the days of non-operation shall take precedence i.e. in the case of doubt, a trip shall be taken as ***not*** operating.

The usage of these within TXC-PTI is as in the table below, described more fully in the sections which follow.

Table 25 - Usage of elements in OperatingProfile

Element Name	Data Type	Used in TXC-PTI
Regular Day Type	*RegularOperationStructure*	Mandated by schema
(one of) DaysOfWeek		
HolidaysOnly	*EmptyType*	Used instead of DaysOfWeek if no regular days of operation are defined.
PeriodicDayType	*PeriodicStructure*	Where appropriate. See section 9.3.3
ServicedOrganisationDayType	*ServicedOrganisationOperation Structure*	Where appropriate. See section 9.3.6
SpecialDaysOperation	*SpecialDaysOperationStructure*	Where appropriate. See section 9.3.4
BankHolidayOperation	*BankHolidayOperationStructure*	Where appropriate. See section 9.3.5

9.3.2 Regular days

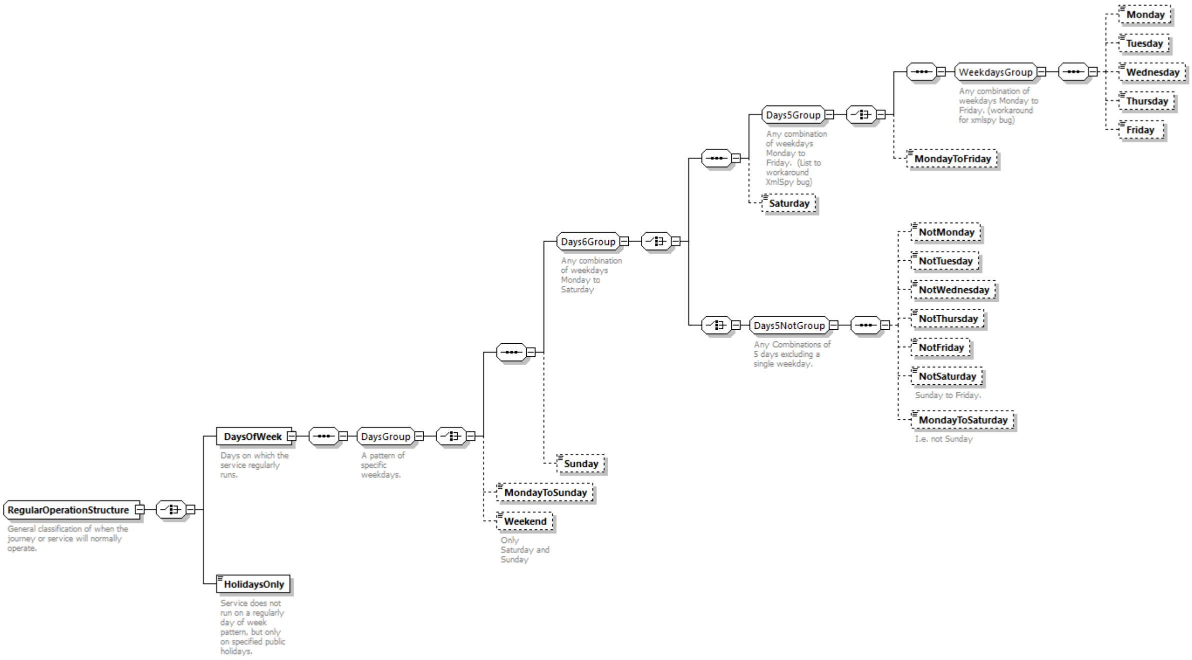

RegularOperationStructure
General classification of when the journey or service will normally operate.
DaysOfWeek
Days on which the service regularly runs.
HolidaysOnly
Service does not run on a regularly day of week pattern, but only on specified public holidays.
DaysGroup
A pattern of specific weekdays.
Days6Group
Any combination of weekdays Monday to Saturday
Sunday
MondayToSunday
Weekend
Only Saturday and Sunday
Days5Group
Any combination of weekdays Monday to Friday. (List to workaround XmlSpy bug)
Saturday
WeekdaysGroup
Any combination of weekdays Monday to Friday. (workaround for xmlspy bug)
Monday
Tuesday
Wednesday
Thursday
Friday
MondayToFriday
Days5NotGroup
Any Combinations of 5 days excluding a single weekday.
NotMonday
NotTuesday
NotWednesday
NotThursday
NotFriday
NotSaturday
Sunday to Friday.
MondayToSaturday
I.e. not Sunday

Figure 27 - The RegularOperationStructure

In TXC, there is an expectation that some degree of regular operation will be defined. Indeed, it is mandated by the schema. There is a degree of confusion as to how the different elements all interact, however, and even more confusion when it comes to the groupings of regular days that can be used as enumerations.

Consider the structure for regular days shown in the figure above. Individual days are clear, as is the intent of some group enumerations such as *MondayToSunday*, *MondayToSaturday* and *MondayToFriday*. However, it becomes much less clear when looking at some of the other grouping enumerations. *NotSaturday*, for example, is annotated with text that describes it as “Sunday to Friday” but as part of a group identified as being a combination of 5 days excluding a single weekday, whereas “weekdays” is a group consisting only of the five individual days of the working week excluding Saturday and Sunday. Furthermore, *NotThursday* (for example), which is unannotated, would suggest a similar structure to *NotSaturday* i.e. Friday to Wednesday. This type of operation seems unlikely.

Because of this potential confusion, and the principle in TXC-PTI to make data clear and specific wherever possible, TXC-PTI shall therefore only use named, individual days of the week and shall not use the group enumerations.

The following example illustrates the use of **RegularDayType** to define a service which operates Mondays, Wednesdays and Fridays.

```
<RegularDayType>
   <!-- Not Tuesdays or Thursdays -->
   <DaysOfWeek>
      <Monday/>
      <Wednesday/>
      <Friday/>
   </DaysOfWeek>
</RegularDayType>
```

Note, particularly, that it is not correct in TXC-PTI to define all days of the week and then remove Tuesdays and Fridays through **SpecialDaysNonOperation**.

The use of a space before the ending / in an enumerated value is allowed within TXC-PTI. For example, either <Monday/> or <Monday /> is valid. It is highly recommended that within a file there is consistency of presentation.

In TXC-PTI, all regular days of operation shall be explicitly stated. Grouping enumerations shall not be used.

If a grouping enumeration is provided the validator will raise an error.

9.3.3 Periodic days

In TXC, periodic days are even less well understood, and consequently are rarely used. These define services which operate on e.g. 1st and 3rd Wednesdays of each month. Such restrictions are often to be found in community transport services taking rural passengers to town market days; these are not their only uses, however.

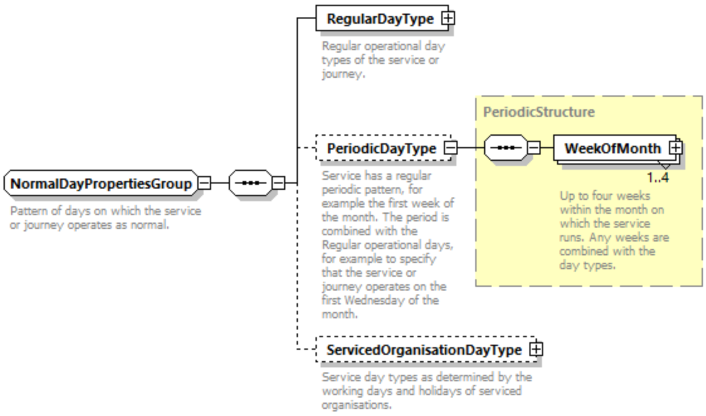

RegularDayType
Regular operational day types of the service or journey.
PeriodicStructure
NormalDayPropertiesGroup
Pattern of days on which the service or journey operates as normal.
PeriodicDayType
WeekOfMonth
1..4
Service has a regular periodic pattern, for example the first week of the month. The period is combined with the Regular operational days, for example to specify that the service or journey operates on the first Wednesday of the month.
Up to four weeks within the month on which the service runs. Any weeks are combined with the day types.
ServicedOrganisationDayType
Service day types as determined by the working days and holidays of serviced organisations.

Figure 28 – The NormalDayPropertiesGroup/PeriodicDayType structure

Traditionally, TXC has tended to encode these services with specific **SpecialDaysOperation** dates. There are, however, two issues with this. The first is that simply having a set of dates does not provide any sort of context as to why those particular dates; the second, potentially more problematic for an operator maintaining data, is that the files have to be continuously refreshed as sets of date ranges expire.

For that reason, in TXC-PTI wherever a trip or service operates on regular, periodic days, then this shall be defined using the **PeriodicDayType** element available in TXC. This element works in conjunction with the **RegularDayType** element: **RegularDayType** states which day(s) of the week the trip or service operates, while **PeriodicDayType** indicates, through a set of week number enumerations, which weeks in that month it operates.

As with all operational day elements, serviced organisations and days of non-operation further qualify these dates – see relevant sections for details.

The previously quoted example, of a service which operates on the 1st and 3rd Wednesdays of each month, would be coded like this in TXC-PTI:

```
<RegularDayType>
   <!—Wednesdays only -->
   <DaysOfWeek>
      <Wednesday/>
   </DaysOfWeek>
</RegularDayType>
<PeriodicDayType>
   <!-- Repeated WeekOfMonth elements, one for each week of operation -->
   <!-- Here,first and third so that overall effect is 1st & 3rd Wednesdays -->
   <WeekOfMonth>
      <WeekNumber>first</WeekNumber>
   </WeekOfMonth>
   <WeekOfMonth>
      <WeekNumber>third</WeekNumber>
   </WeekOfMonth>
</PeriodicDayType>
```

Note that the **WeekNumber** element is an enumeration, and not a number. Possible values are *first*, *second*, *third*, *fourth*, *fifth*, and *last*.

In TXC-PTI, all regular periodic operation shall be coded using the **PeriodicDayType** element, and not with individual date ranges.

The validator tests for the use of enumerated values in WeekOfMonth and WeekNumber, providing an integer will trigger an error in validation.

9.3.4 Special days

Special days are the method by which additional regular, irregular, or one-off dates or exclusions can be set up for trips. Such trips can include festival services, shuttle services which only operate when points of interest are open, football services, and similar.

Special days are structured with two distinct sections: **DaysOfOperation**, describing the additional dates that the trip or service operates; and **DaysOfNonOperation**, describing the dates when the service does not operate. In both cases a series of date ranges is included, with inclusive **StartDate** and **EndDate** elements and a **Note** if required.

Note that, in TXC and consequently also in TXC-PTI, special days **DaysOfOperation** are *additive*; that is they operate in addition to any other stated days. This is true even if the special day occurs on a different day of the week to the stated regular days. This is particularly important to note if a range of dates is defined, since the trip or service would operate on *every single day within that date range*.

Where a trip or service is only operating on a particular set of dates, and does not have a regular day type, the coding **HolidaysOnly** shall be used in **RegularDayType** in place of **DaysOfWeek**.

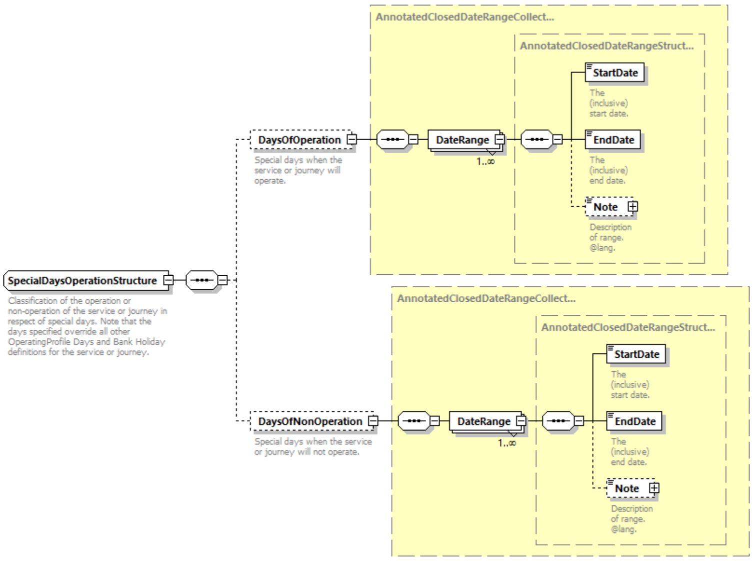

AnnotatedClosedDateRangeCollect...
AnnotatedClosedDateRangeStruct...
StartDate
The (inclusive) start date.
DaysOfOperation
Special days when the service or journey will operate.
DateRange
1..∞
EndDate
The (inclusive) end date.
Note
Description of range. @lang.
SpecialDaysOperationStructure
Classification of the operation or non-operation of the service or journey in respect of special days. Note that the days specified override all other OperatingProfile Days and Bank Holiday definitions for the service or journey.
AnnotatedClosedDateRangeCollect...
AnnotatedClosedDateRangeStruct...
StartDate
The (inclusive) start date.
DaysOfNonOperation
Special days when the service or journey will not operate.
DateRange
1..∞
EndDate
The (inclusive) end date.
Note
Description of range. @lang.

Figure 29 - The structure of special days

For example, the following code would define a trip which operated only during a four-day event:

```
<OperatingProfile>
     <RegularDayType>
        <!—No regular operation -->
        <HolidaysOnly/>
     </RegularDayType>
     <SpecialDaysOperation>
        <DaysOfOperation>
           <!-- Festival operation – trip operates on all dates in range -->
           <DateRange>
              <StartDate>2020-07-30</StartDate>
              <EndDate>2020-08-02</EndDate>
           </DateRange>
        </DaysOfOperation>
     </SpecialDaysOperation>
</OperatingProfile>
```

On the contrary, **DaysOfNonOperation** are *restrictive* (or *subtractive*); that is, they restrict the already stated days of operation. These could cover, for example, one-off events such as marathons which take place on known dates, but which temporarily disrupt services. For example:

```
<OperatingProfile>
     <RegularDayType>
        <!-- Sunday operation -->
        <DaysOfWeek>
           <Sunday/>
        </DaysOfWeek>
     </RegularDayType>
     <SpecialDaysOperation>
        <DaysOfNonOperation>
           <!-- Great North Run 2020 -->
           <DateRange>
              <StartDate>2020-09-13</StartDate>
```

```
          <EndDate>2020-09-13</EndDate>
        </DateRange>
      </DaysOfNonOperation>
    </SpecialDaysOperation>
</OperatingProfile>
```

The general principle in TXC-PTI is that dated ranges should only be used as a last resort. Often, trips are better coded, and contain more contextual information, using regular day types, periodic days, named bank holidays (whether using built-in enumerations or especially created) or serviced organisations. See other sections for more detail.

In TXC-PTI, special operating dates shall be used only as a last resort when all other options have been explored.

In the operating profile, if SpecialDaysOperation is used but no other day type (RegularDayType, PeriodicDayType or ServicedOrganisationDayType) is provided then a warning is raised.

9.3.5 Bank holidays

Bank holiday enumerations have proved to be problematic to some users, particularly those in Scotland. This is because systems have tended to use groups of bank holidays, such as *AllBankHolidays* to simplify the output. In certain jurisdictions, though, these groups contain bank holidays with assigned dates that are not universally recognised. This causes normal operations to drop out of systems unexpectedly and results in significant work to correct.

One solution that was proposed is to dispense with the encoded bank holidays entirely and to rely on dated ranges to state when trips did and did not run. However, this is to throw out a significant degree of context provided by e.g. “does not run on Good Friday”.

An alternative solution proposed using facilities in TXC to define only those bank holidays that were required, using special date ranges. However, this option was also discounted as it could lead to conflicts when named bank holidays were created for holidays that were already enumerated.

TXC-PTI has therefore adopted the principle that grouping elements shall not be used, and all bank holidays shall be explicitly stated in their appropriate **DaysOfOperation** or **DaysofNonOperation** structures. This is in line with the principles already discussed for regular days of operation above, and allows contextual bank holiday operation to be provided while removing problematic bank holidays from consideration.

Within the TXC bank holiday enumeration there are two days that are not strictly bank holidays – ChristmasEve and NewYearsEve. The for the purposes of TXC and the TXC-PTI these days are to be regarded as bank holidays and should be coded as such.

Note that nothing in the foregoing prevents additional bank holidays being defined using the **OtherPublicHoliday** element, as some special days are introduced from time to time such as the Royal Wedding and Queen’s Jubilee additional days in 2012. For the structure of the **OtherPublicHoliday** element, please refer to the TXC schema guidance.

For an example of bank holiday coding, consider the following snippet of XML. As structured, this would prevent a trip from operating Christmas Day, Boxing Day, New Year’s Day, the 2nd January holiday in Scotland, and Easter Monday, but would allow it to continue running on St Andrews Day and a Scottish summer bank holiday, neither of which are routinely observed.

```
<BankHolidayOperation>
  <DaysOfNonOperation>
    <ChristmasDay/>
    <BoxingDay/>
    <NewYearsDay/>
    <Jan2ndScotland/>
    <EasterMonday/>
  </DaysOfNonOperation>
</BankHolidayOperation>
```

In TXC-PTI, all bank holidays with enumerations shall be explicitly stated within the **DaysOfOperation** and/or **DaysOfNonOperation** blocks. Grouping enumerations shall not be used.

For services that operate wholly within England and Wales the following are the bank holidays that must be explicitly coded either as DaysOfOperation or DaysOfNonOperation:

- ChristmasEve
- NewYearsEve
- ChristmasDay
- ChristmasDayHoliday
- BoxingDay
- BoxingDayHoliday
- NewYearsDay
- NewYearsDayHoliday
- GoodFriday
- EasterMonday
- MayDay
- SpringBank
- LateSummerBankHolidayNotScotland

The coding of Scottish bank holidays is not mandatory, services with one or more stops in Scotland are recommended to code the additional days to ensure that Scottish data consumers receive accurate data:

- Jan2ndScotland
- Jan2ndScotlandHoliday
- StAndrewsDay
- StAndrewsDayHoliday
- AugustBankHolidayScotland

Where an operator with a service that runs entirely within Scotland wishes to submit data to BODS then all the English bank holidays will need to be fully populated for the service to pass the validation tests.

As a result of the approach that TXC takes to specify bank holidays and movable day of the week for some bank holidays, for example Boxing Day and New Year's day it is important that the TXC files submitted to BODS reflect the correct operation from year to year. It is probable that even if the timetable has not changed between years that TXC files will need to be re-submitted to ensure that bank holiday and DaysOfOperation data are correctly supplied.

Note that, in TXC and consequently also in TXC-PTI, bank holiday **DaysOfOperation** are *additive*; that is they operate in addition to any other stated days. This is true even if the bank holiday occurs on a different day of the week to the stated regular days. For example, the following code would define a trip which operated “Sundays and Bank Holidays (excluding Christmas and New Year)” in England:

```
<OperatingProfile>
     <RegularDayType>
        <!-- Sunday operation -->
        <DaysOfWeek>
           <Sunday/>
        </DaysOfWeek>
     </RegularDayType>
     <BankHolidayOperation>
        <DaysOfOperation>
           <!-- Additional bank holiday operation -->
           <GoodFriday/>
           <LateSummerBankHolidayNotScotland/>
           <MayDay/>
           <EasterMonday/>
           <SpringBank/>
        </DaysOfOperation>
        <DaysOfNonOperation>
           <!-- Bank holiday non-operation -->
           <ChristmasEve/>
           <ChristmasDay/>
           <ChristmasDayHoliday/>
           <BoxingDay/>
           <BoxingDayHoliday>
           <NewYearsEve/>
           <NewYearsDay/>
           <NewYearsDayHoliday/>
        </DaysOfNonOperation>
     </BankHolidayOperation>
</OperatingProfile>
```

On the contrary, **DaysOfNonOperation** are *restrictive* (or *subtractive*); that is, they restrict the already stated days of operation. For example, a trip which is Mondays to Fridays excluding bank holidays would be coded as shown below.

```
<OperatingProfile>
     <RegularDayType>
        <!-- Weekday operation -->
        <DaysOfWeek>
           <Monday/>
           <Tuesday/>
           <Wednesday/>
           <Thursday/>
           <Friday/>
        </DaysOfWeek>
     </RegularDayType>
     <BankHolidayOperation>
        <DaysOfNonOperation>
           <!-- Bank holiday non-operation -->
           <ChristmasEve/>
           <ChristmasDay/>
           <ChristmasDayHoliday/>
           <BoxingDay/>
           <BoxingDayHoliday>
```

```
                   <NewYearsEve/>
                   <NewYearsDay/>
                   <NewYearsDayHoliday/><GoodFriday/>
                   <LateSummerBankHolidayNotScotland/>
                   <MayDay/>
                   <EasterMonday/>
                   <SpringBank/>
                </DaysOfNonOperation>
             </BankHolidayOperation>
        </OperatingProfile>
```

9.3.6 Referencing serviced organisations

The coding and the application of serviced organisations has already been covered in section 3 above, and reference was made there to the way in which they are encoded in TXC-PTI **VehicleJourneys**.

To reiterate the fundamental point made in that earlier section, in TXC-PTI a serviced organisation shall only be provided with dates for working days, and all other dates shall be assumed to be holidays. This means that trips which operate on working (e.g. school) days only are coded as **DaysOfOperation/WorkingDays**, while trips which only operate during the holidays will be encoded as **DaysOfNonOperation/WorkingDays**.

An additional consideration for using serviced organisation dates within a TXC-PTI file is the manner in which the **ServicedOrganisation** interacts with the regular days. This is something which causes a degree of confusion in TXC, and it is important to note that serviced organisation dates *restrict* (are logically ANDed with) the days of the week provided in the **RegularDayType** element. So, for schools, for example, which by and large operate Mondays to Fridays, this pattern can be stated as the regular days of operation. The dates provided for the serviced organisation then list the range of dates during which those Mondays to Fridays apply.

Thus, there is no need for a **ServicedOrganisation** to define an entire half-term's worth of Monday to Friday dates. It is far simpler to define a single date range for the half term (e.g. 5 September – 18 October) and use the **RegularDayType** to limit this to Mondays to Fridays within that date range. This is much simpler to code and to interpret / debug.

This principle also carries forward into bank holidays. For the most part, bank holidays usually fall within school holidays. However, depending on the date of Easter and the Early May Holiday, some bank holidays can occur during term time. Given that dates of non-operation further limit the standard operation, it follows that there is no need to exclude bank holidays from the range of working days.

For example:

```
     <ServicedOrganisation>
        <OrganisationCode>HPFLD</OrganisationCode>
        <Name>Harrop Fold School</Name>
        <WorkingDays>
        <DateRange>
             <StartDate>2020-02-24</StartDate>
             <EndDate>2020-04-03</EndDate>
             <Description>Spring Term 2020 - second half</Description>
        </DateRange>
        <DateRange>
             <!-- Date range omits 9 May bank holiday -->
             <StartDate>2020-04-20</StartDate>
             <EndDate>2020-05-22</EndDate>
             <Description>Summer Term 2020 - first half</Description>
        </DateRange>
        <DateRange>
             <StartDate>2020-06-01</StartDate>
             <EndDate>2020-07-16</EndDate>
             <Description>Summer Term 2020 – second half</Description>
        </DateRange>
        </WorkingDays>
     </ServicedOrganisation>
     ...
     <VehicleJourneys>
        <VehicleJourney SequenceNumber="1">
        <OperatingProfile>
             <RegularDayType>
               <!-- Weekday operation -->
               <DaysOfWeek>
                  <Monday/>
                  <Tuesday/>
                  <Wednesday/>
                  <Thursday/>
                  <Friday/>
               </DaysOfWeek>
             </RegularDayType>
             <ServicedOrganisationDayType>
               <DaysOfOperation>
                  <!-- Weekdays restricted by the dates for organisation HPFLD -->
                  <WorkingDays>
                     <ServicedOrganisationRef>HPFLD</ServicedOrganisationRef>
                  </WorkingDays>
               </DaysOfOperation>
             </ServicedOrganisationDayType>
             <BankHolidayOperation>
               <DaysOfNonOperation>
                  <!-- Bank holiday non-operation further excludes 9 May -->
                  <!-- (MayDay in 2020) -->
                  <ChristmasDay/>
                  <BoxingDay/>
                  <GoodFriday/>
                  <NewYearsDay/>
                  <LateSummerBankHolidayNotScotland/>
                  <MayDay/>
                  <EasterMonday/>
                  <SpringBank/>
               </DaysOfNonOperation>
             </BankHolidayOperation>
        </OperatingProfile>
        ...
        </VehicleJourney>
        ...
     </VehicleJourneys>
```

In TXC-PTI, serviced organisation dates restrict the regular day types, and are themselves further restricted by bank holiday non-operation. Thus, working date ranges for serviced organisations shall be stated simply.

9.4 Trip timings

As has previously been stated, the assumption in TXC-PTI is that information relating to a trip shall be fully defined within a trip. However, for reasons of compactness and readability it can be desirable, at times, to define some of this information at a higher level. TXC-PTI therefore allows this to happen, so long as any difference in a **VehicleJourney** is indicated by a complete replacement of the relevant blocks.

This is true for times at stop (or, more precisely, the times over the links in the vehicle journey). As described in section 8, a **JourneyPattern** can be defined with run times of zero minutes over each link, with the actual times specified in each **VehicleJourney**. This is accomplished using a series of **VehicleJourneyTimingLink** elements, with the link times provided in the **RunTime** element, and the **WaitTime** (if any) defined in both the **To** stop at the end of one link and the **From** stop at the start of the next link.

A **VehicleJourneyTimingLink** has a similar – but not identical – set of elements a **JourneyPatternTimingLink**, and it is instructive to compare the structure shown below with Figure 23 in section 8.4. It can be seen that:

- A **VehicleJourneyTimingLink** has a reference back to the original **JourneyPatternTimingLink** via the **JourneyPatternTimingLinkRef** element.
- The **From** and **To** elements do not refer to the stop since this is inherited from the referenced journey pattern.
- In addition, for TXC-PTI the stop activity shall also be inherited from the journey pattern and not repeated in the vehicle journey. This is because, in general, it is unlikely that a stop will change whether it is set down or not (etc.) between trips. Note that timing point status is always inherited as there is no mechanism to override it in TXC.
- The common timing link group is common to both **VehicleJourneyTimingLink** and **JourneyPatternTimingLink**, and the same rules apply.

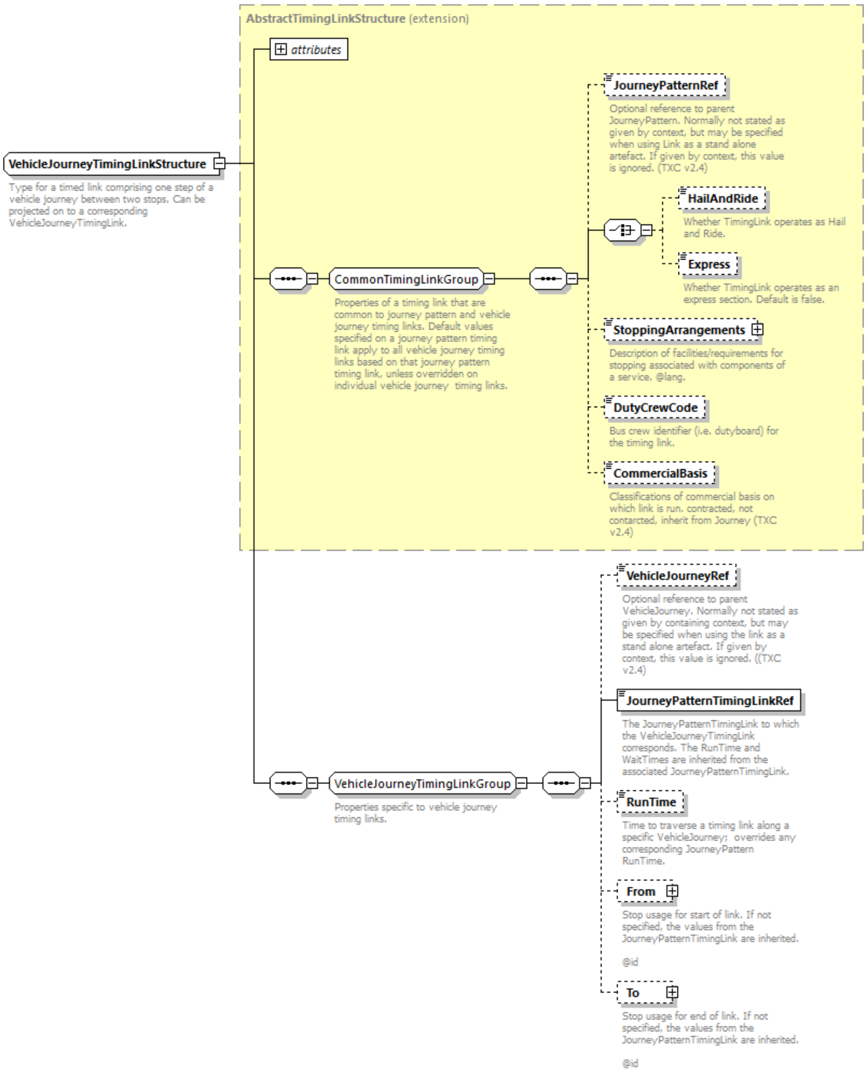

AbstractTimingLinkStructure (extension)
attributes
JourneyPatternRef
Optional reference to parent JourneyPattern. Normally not stated as given by context, but may be specified when using Link as a stand alone artefact. If given by context, this value is ignored. (TXC v2.4)
VehicleJourneyTimingLinkStructure
Type for a timed link comprising one step of a vehicle journey between two stops. Can be projected on to a corresponding VehicleJourneyTimingLink.
HailAndRide
Whether TimingLink operates as Hail and Ride.
CommonTimingLinkGroup
Properties of a timing link that are common to journey pattern and vehicle journey timing links. Default values specified on a journey pattern timing link apply to all vehicle journey timing links based on that journey pattern timing link, unless overridden on individual vehicle journey timing links.
Express
Whether TimingLink operates as an express section. Default is false.
StoppingArrangements
Description of facilities/requirements for stopping associated with components of a service. @lang.
DutyCrewCode
Bus crew identifier (i.e. dutyboard) for the timing link.
CommercialBasis
Classifications of commercial basis on which link is run. contracted, not contarcted, inherit from Journey (TXC v2.4)
VehicleJourneyRef
Optional reference to parent VehicleJourney. Normally not stated as given by containing context, but may be specified when using the link as a stand alone artefact. If given by context, this value is ignored. ((TXC v2.4)
JourneyPatternTimingLinkRef
The JourneyPatternTimingLink to which the VehicleJourneyTimingLink corresponds. The RunTime and WaitTimes are inherited from the associated JourneyPatternTimingLink.
VehicleJourneyTimingLinkGroup
Properties specific to vehicle journey timing links.
RunTime
Time to traverse a timing link along a specific VehicleJourney; overrides any corresponding JourneyPattern RunTime.
From
Stop usage for start of link. If not specified, the values from the JourneyPatternTimingLink are inherited.
@id
To
Stop usage for end of link. If not specified, the values from the JourneyPatternTimingLink are inherited.
@id

Figure 30 - The VehicleJourneyTimingLink structure

It should be noted that a **VehicleJourney** does not have the same Pattern > Section > Link structure as a **JourneyPattern**; **VehicleJourneyTimingLink** elements form a continuous sequence. For TXC-PTI, there can be no missing links in this sequence – there must be a one-to-one mapping between the links in a set of **VehicleJourneyTimingLink** element and the underlying **JourneyPattern** that is referenced by the **VehicleJourney**. If any links are to be omitted, then a corresponding **JourneyPattern** omitting the same links must exist.

The table below shows the permitted elements for the VehicleJourneyTimingLink.

Table 26 - Permitted elements in VehicleJourneyTimingLink

Element Name	**Data Type**	**Used in TXC-PTI**
JourneyPatternRef	*JourneyPatternRefStructure*	Not used
HailAndRide	*boolean*	Not used
Express	*boolean*	Not used
StoppingArrangements	*NaturalLanguageString*	Not used
DutyCrewCode	*DutyCrewCodeType*	Optional
CommercialBasis	*LinkCommercialBasisEnumeration*	Not used at this level. Use within Service or VehicleJourney. See sections 5.3.3. and 9.3
VehicleJourneyRef	*VehicleJourneyCodeType*	Optional
JourneyPatternTimingLinkRef	*JourneyPatternTimingLinkRef Structure*	Mandated by schema
RunTime	*DurationType*	Yes, shall be included and shall be the actual run time.
From	*VehicleJourneyStopUsageStructure*	May be included in order to provide a **WaitTime** or a different **DynamicDestinationDisplay**. Other elements shall not be used. **WaitTime** shall match the WaitTime on the preceding **To** link
To	*VehicleJourneyStopUsageStructure*	May be included in order to provide a **WaitTime** or a different **DynamicDestinationDisplay**. Other elements shall not be used. **WaitTime** shall match the WaitTime on the following **From** link

In TXC-PTI, where a default pattern is provided for timing links then this can be overridden by use of **VehicleJourneyTimingLink**. Where this is used then there shall be a one-to-one mapping to a **JourneyPattern** with the same set of links, and times shall be provided for all links and (where necessary) wait times. No links shall be omitted.

9.5 Journeys after midnight

TXC is a calendar-based system, and trips are defined as running on the actual calendar day on which they operate. However, many operators use an “operational day” concept, with days which start at (for example) 03:00 and run until 02:59 the following day.

For many services this leads to trips which run after midnight being shown at the end of a timetable panel, belonging to the previous calendar day. For example, trips in larger towns and cities, especially those with a sizeable student population, will often run trips in the early hours of Saturday morning and Sunday morning. These are shown, respectively, at the end of a Monday to Friday panel, labelled “Fridays only” and at the end of a Saturday panel.

These types of journeys are often wrongly coded in TXC, or are coded in a way that makes it difficult for a consuming system to place them in the right place when presenting the information in a traditional timetable. For example, TXC exists where:

- trips are shown with the incorrect day type.

  In the example above, this would result in a trip that should operate in the early hours of Saturday morning being shown with a “Friday” operating date meaning that is being described, incorrectly, as operating in the early hours of Friday morning.

- trips are shown with the correct day type but cause unnecessary complexity on timetables.

  This is particularly true on a Monday, where trips on Sunday night into Monday are shown alongside similar, but differently timed, trips on other weekdays.

A number of different mechanisms were implemented in TXC v2.4 in order to overcome this limitation, some of which are more complicated than others. For example, there is scope for an operator to declare particular day types of their own and to map regular days and times into them. For TXC-PTI, though, this was considered too complicated and the aim was to have something that was simple to grasp, and not that different from the standardised, TXC v2.1, way of encoding data.

To that end, TXC-PTI trips which run after midnight shall use the **DepartureDayShift** coding. This is an offset that allows a trip to be coded to the correct operational day, but then given an offset as to when it actually runs. For TXC-PTI this shall only ever be “+1”. Shifting departures backwards, or forwards by a greater amount, shall not be permitted.

For example:

```
<VehicleJourney>
  <!-- Trip runs operationally on Sunday -->
  <!-- Trip shall be displayed at the end of the Sunday timetable -->
  <!-- Trip physically runs at 00:15 on a Monday morning -->
  <OperatingProfile>
    <RegularDayType>
      <DaysOfWeek>
        <Sunday>
      </DaysOfWeek>
    </RegularDayType>
  </OperatingProfile>
  <VehicleJourneyCode>VJ1</VehicleJourneyCode>
  <ServiceRef>S1</ServiceRef>
  <LineRef>L1</LineRef>
  <JourneyPatternRef>JP1</JourneyPatternRef>
  <DepartureTime>00:15:00</DepartureTime>
  <DepartureDayShift>+1</DepartureDayShift>
</VehicleJourney>
```

and:

```
<VehicleJourney>
  <!-- Trip runs operationally on weekdays -->
```

```
   <!-- Trip shall be displayed at end of the Monday to Friday timetable -->
   <!-- Trip physically runs at 00:25 on Tuesday to Saturday mornings -->
   <OperatingProfile>
      <RegularDayType>
         <DaysOfWeek>
            <Monday/>
            <Tuesday/>
            <Wednesday/>
            <Thursday/>
            <Friday/>
         </DaysOfWeek>
      </RegularDayType>
   </OperatingProfile>
   <VehicleJourneyCode>VJ1</VehicleJourneyCode>
   <ServiceRef>S1</ServiceRef>
   <LineRef>L1</LineRef>
   <JourneyPatternRef>JP1</JourneyPatternRef>
   <DepartureTime>00:25:00</DepartureTime>
   <DepartureDayShift>+1</DepartureDayShift>
</VehicleJourney>
```

In TXC-PTI, operational days shall be described through use of **DepartureDayShift** to code trips to the correct operational and calendar days.

When DepartureDayShift is used only a shift of one day (1) is allowed.

9.6 Interchanges between vehicle journeys

9.6.1 Basic structure of a VehicleJourneyInterchange

As discussed in section 5.3.6, interchanges are a key piece of information, and shall be included in TXC-PTI wherever possible. That section discusses how service-level interchanges are described, interchanging between journey patterns. However, this is not always possible – for example, not all trips connect - and there may be occasions when interchanges need to be described at vehicle journey level. This is achieved with a **VehicleJourneyInterchange**.

A **VehicleJourneyInterchange** is constructed in a very similar manner to the **JourneyPatternInterchange**, and it is instructive to compare the structure shown in Figure 31 on the following page with Figure 13 in section 5.3.6.

To start with the *CommonInterchangeGroup* is the same in both, and the **VehicleJourneyInterchange** elements within that shall therefore follow the same rules as set out for the **JourneyPatternInterchange** in section 5.3.6.2 and in Table 12.

Beyond that, it is simply a case of specifying the incoming and outgoing vehicle journeys in place of the incoming and outgoing journey patterns, using **InboundVehicleJourneyRef** and **OutboundVehicleJourneyRef** elements, and specifying the stops at which the connection occurs using **InboundStopPointRef** and **OutboundStopPointRef**. All four elements are mandatory for TXC-PTI.

It is recommended that, in TXC-PTI, interchanges which can be made between trips on the same service, but which cannot be specified at service level, should be included using **VehicleJourneyInterchange** elements as described above.

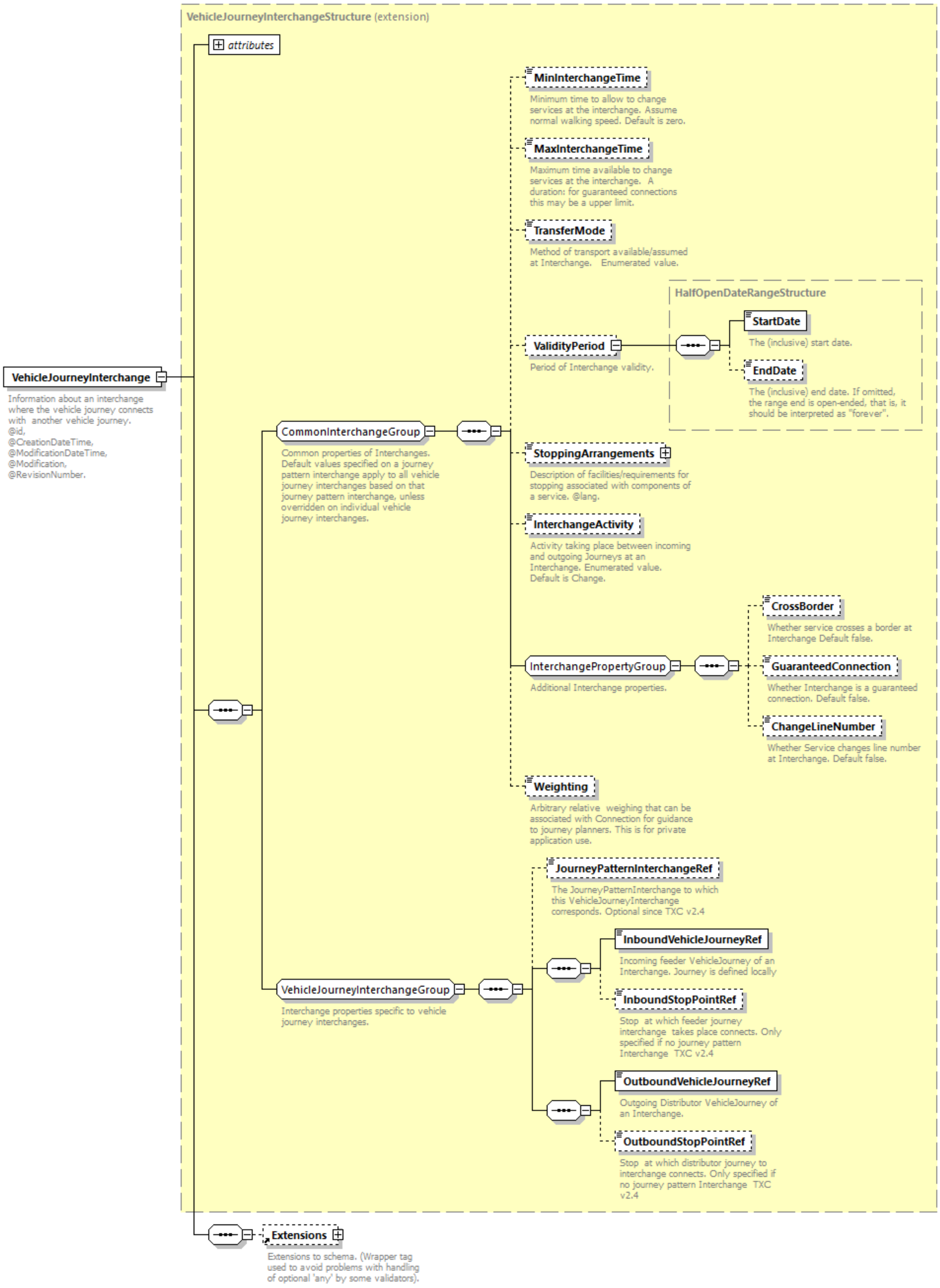

VehicleJourneyInterchangeStructure (extension)
attributes
MinInterchangeTime
Minimum time to allow to change services at the interchange. Assume normal walking speed. Default is zero.
MaxInterchangeTime
Maximum time available to change services at the interchange. A duration: for guaranteed connections this may be a upper limit.
TransferMode
Method of transport available/assumed at Interchange. Enumerated value.
HalfOpenDateRangeStructure
StartDate
The (inclusive) start date.
ValidityPeriod
Period of Interchange validity.
EndDate
The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever".
VehicleJourneyInterchange
Information about an interchange where the vehicle journey connects with another vehicle journey. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
CommonInterchangeGroup
Common properties of Interchanges. Default values specified on a journey pattern interchange apply to all vehicle journey interchanges based on that journey pattern interchange, unless overridden on individual vehicle journey interchanges.
StoppingArrangements
Description of facilities/requirements for stopping associated with components of a service. @lang.
InterchangeActivity
Activity taking place between incoming and outgoing Journeys at an Interchange. Enumerated value. Default is Change.
CrossBorder
Whether service crosses a border at Interchange Default false.
InterchangePropertyGroup
Additional Interchange properties.
GuaranteedConnection
Whether Interchange is a guaranteed connection. Default false.
ChangeLineNumber
Whether Service changes line number at Interchange. Default false.
Weighting
Arbitrary relative weighing that can be associated with Connection for guidance to journey planners. This is for private application use.
JourneyPatternInterchangeRef
The JourneyPatternInterchange to which this VehicleJourneyInterchange corresponds. Optional since TXC v2.4
InboundVehicleJourneyRef
Incoming feeder VehicleJourney of an Interchange. Journey is defined locally
VehicleJourneyInterchangeGroup
Interchange properties specific to vehicle journey interchanges.
InboundStopPointRef
Stop at which feeder journey interchange takes place connects. Only specified if no journey pattern Interchange TXC v2.4
OutboundVehicleJourneyRef
Outgoing Distributor VehicleJourney of an Interchange.
OutboundStopPointRef
Stop at which distributor journey to interchange connects. Only specified if no journey pattern Interchange TXC v2.4
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 31 - The VehicleJourneyInterchange element

The following table shows the usage of the elements within a **VehicleJourneyInterchange** in TXC-PTI documents.

Table 27 - Usage of elements in a VehicleJourneyInterchange

Element Name		**Data Type**	**Used in TXC-PTI**
MinInterchangeTime		*DurationType*	The minimum permitted interchange time (default = 0 mins). Use where appropriate
MaxInterchangeTime		*DurationType*	The maximum time for a guaranteed connection. Use where appropriate.
TransferMode		*AllModesEnumeration*	Not used as assumed that the interchange mode is the mode of the service.
ValidityPeriod	StartDate	*StartDateType*	Not used as assumed that the interchange is valid for the duration of the **OperatingPeriod**.
	EndDate	*EndDateType*	
StoppingArrangements		*NaturalLanguageString*	Not used.
InterchangeActivity		*InterchangeActivity Enumeration*	Mandated in TXC-PTI interchanges. Describes the interchange as follows: *transferOnly* not used. *change* used for change of vehicle. *through* used to stay on vehicle. *split* not used. *join* not used.
CrossBorder		*boolean*	Optional.
GuaranteedConnection		*boolean*	Mandated in TXC-PTI interchanges. Set to true for a guaranteed connection which require changes of vehicle, or for through connections with the same line name. False otherwise.
ChangeLineNumber		*boolean*	Mandated in TXC-PTI interchanges. Set to true where a through connection changes line name. False otherwise.
Weighting		*integer*	Optional.
JourneyPatternInterchangeRef		*JourneyPatternInterchange IdType*	Optional.
InboundVehicleJourneyRef		*VehicleJourneyRef Structure*	Mandated by schema. Reference to the feeder vehicle journey
InboundStopPointRef		*StopPointRefStructure*	Yes, shall be included. Reference to the stop point at which the interchange takes place.
OutboundVehicleJourneyRef		*VehicleJourneyCodeType*	Mandated by schema. Reference to the distrubutor vehicle journey
OutboundStopPointRef		*StopPointRefStructure*	Yes, shall be included. Reference to the stop point at which the interchange takes place.

9.6.2 Example of a vehicle journey interchange

The following example shows how an interchange might be defined between two trips at a vehicle journey level. For comparison, the same First in Leicester lines 16 from the example of the journey pattern interchange in section 5.3.6.3 has been used.

Line 16 has connections between two different vehicle journeys in different directions, connecting at NaPTAN stop reference 269039050. As before, note here that the direction of the line implied by the vehicle journey reference (showing OB for outbound and IB for inbound) is *not* the same as the usage of those terms in the interchange. Inbound in the interchange is for the incoming, feeder trip while outbound is for the outgoing, distributor trip.

```
<VehicleJourneyInterchange>
   <MinInterchangeTime>PT0M</MinInterchangeTime>
   <InterchangeActivity>through</InterchangeActivity>
   <GuaranteedConnection>true</GuaranteedConnection>
   <InboundVehicleJourneyRef>VJ1-MF-OB-1</InboundVehicleJourneyRef>
   <InboundStopPointRef>269039050</InboundStopPointRef>
   <OutboundVehicleJourneyRef>VJ17-MF-IB-2</OutboundVehicleJourneyRef>
   <OutboundStopPointRef>269039050</OutboundStopPointRef>
</VehicleJourneyInterchange>
```

9.7 Real-time information

9.7.1 Introduction

In a TXC-PTI document, information to support real time information shall be included where available. While this is not public-facing information, and TXC-PTI has been designed around information that is directly relevant to the public, these elements are nevertheless an important part of the information feeding real time systems.

There are undoubted tensions between the need to provide timely notice of timetable changes and the (generally much later) allocation of vehicle duty schedules to the timetables. For this reason, the real time information is not mandatory in TXC-PTI, although the prompt, early inclusion of this information is strongly recommended.

For the avoidance of doubt, TXC-PTI will mandate that, where this information is provided, it shall use the same references for the equivalent elements that are used in SIRI-VM feeds.

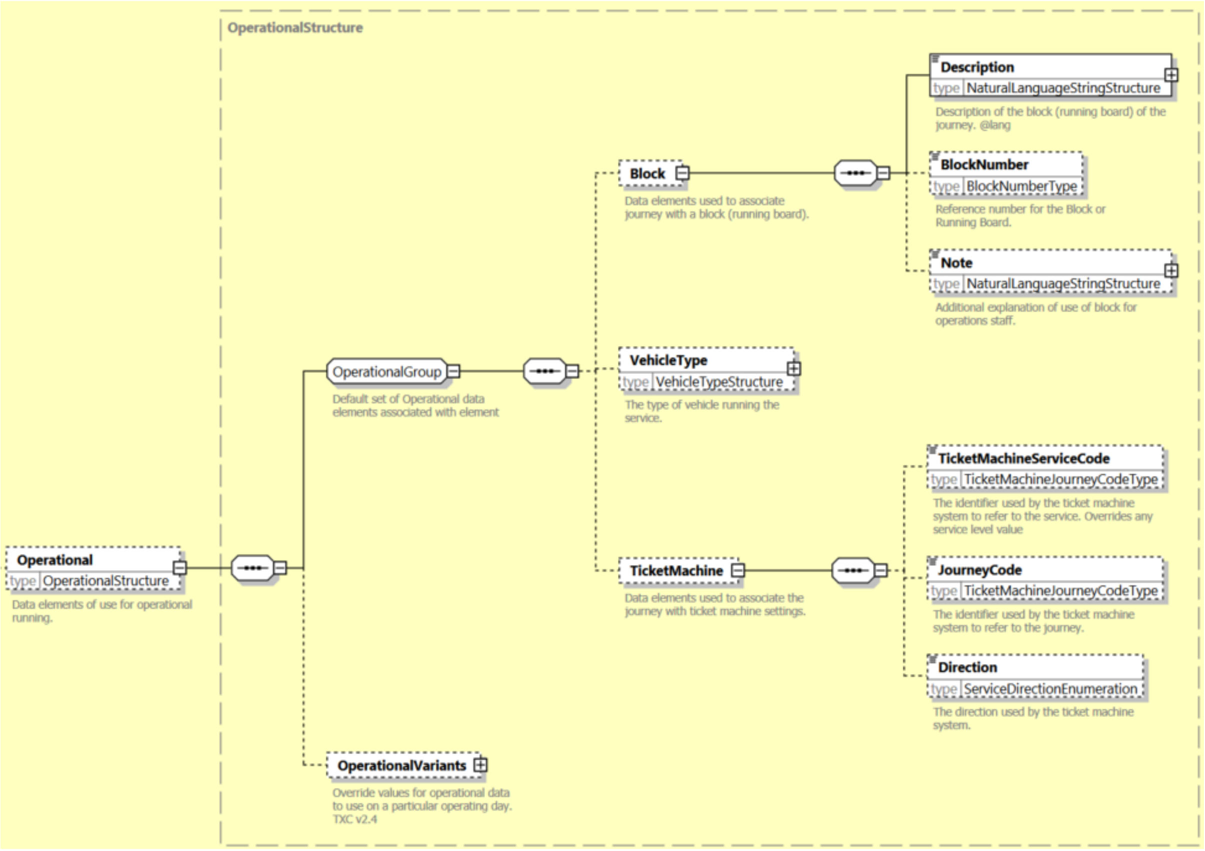

OperationalStructure
Operational
type OperationalStructure
Data elements of use for operational running.
OperationalGroup
Default set of Operational data elements associated with element
Block
Data elements used to associate journey with a block (running board).
Description
type NaturalLanguageStringStructure
Description of the block (running board) of the journey. @lang
BlockNumber
type BlockNumberType
Reference number for the Block or Running Board.
Note
type NaturalLanguageStringStructure
Additional explanation of use of block for operations staff.
VehicleType
type VehicleTypeStructure
The type of vehicle running the service.
TicketMachine
Data elements used to associate the journey with ticket machine settings.
TicketMachineServiceCode
type TicketMachineJourneyCodeType
The identifier used by the ticket machine system to refer to the service. Overrides any service level value
JourneyCode
type TicketMachineJourneyCodeType
The identifier used by the ticket machine system to refer to the journey.
Direction
type ServiceDirectionEnumeration
The direction used by the ticket machine system.
OperationalVariants
Override values for operational data to use on a particular operating day. TXC v2.4

Figure 32 - The OperationalStructure supporting real time elements

Elements that may be populated in TXC-PTI include **LayoverPoint** and **GarageRef**, together with their linked dead runs (**StartDeadRun** and **EndDeadRun**), and the **Block BlockNumber** and **TicketMachine JourneyCode** elements within **Operational** element.

OperationalVariants, however, shall not be included. Where there are variants, these are best described using specific vehicle journeys.

In common with the principle of explicit declarations, where these are included, they shall be fully declared as part of a **VehicleJourney**, and not elsewhere.

JourneyCode and **Block** shall be provided as soon as they are available from operational management systems.

JourneyCode must be unique within the Line. There is no fixed format for **JourneyCode** to allow for the different approaches used by operators and system suppliers. Typically it will be populated with data that a driver would enter into their ticket machine, for example a journey number or journey start time.

```
<TicketMachine>
   …
   <JourneyCode>43</JourneyCode>
</TicketMachine>

<TicketMachine>
   …
   <JourneyCode>1633</JourneyCode>
</TicketMachine>
```

JourneyCode will be used to match against VehicleJourneyRef within the location data feeds.

BlockNumber is also known as bus workings number, most frequently populated using running board information. It's a unique identifier or code (usually a simple number) that is used for all the journeys an individual bus is scheduled to work. Examples of BlockNumber include:

```
<Block>
   <BlockNumber>203</BlockNumber>
</Block>

<Block>
   <BlockNumber>115106</BlockNumber>
</Block>
```

BlockNumber will be used to match against BlockRef within the location data feeds.

It is recommended that, in TXC-PTI, real time elements are provided as soon as they become available.

Where real time information is provided, references in TXC-PTI shall match with equivalent references in SIRI-VM feeds.

The provision of JourneyCode is important for data consumers to be able to use in order to match journeys between the TXC-PTI and SIR VM location data provided to BODS. This enables data consumers to provide accurate journey predictions for the current vehicle journey for passengers. JourneyCode from the TXC-PTI shall be provided in the SIRI VM from the location data provider.

The provision of Block is important for data consumers to be able to use in order to match across different data types. This enables data consumers to improve their journey predictions for passengers by considering cross journey predictions. Meaning if a vehicle is late on a journey, the next journey is likely to also be running late.

BODS recognises that Block information may not be available at the time of initial submission to BODS and or registration. The Data Quality reports observation identifies if the service is valid in the next week, and when it is valid it contains block numbers. The block number needs to be the same as the corresponding object in the bus location data field 'BlockRef'.

If an operator does not have a documented running board, they can create one by allocating each of their vehicles a 'code'. For each of the journeys operated by the same vehicle, the journey should be given a consistent identifier as the input for both the TransXChange (BlockNumber) and SIRI-VM (BlockRef). For example, Vehicle 1, could have block number = 1 allocated to all the journeys that will be completed by vehicle 1.

9.7.2 Layover points

Layover points specify places, normally not bus stops, where buses will wait between trips. They also specify for how long the bus will remain at that location. They are useful in real time information systems to predict recovery times, delays to follow-on trips, etc.

A **LayoverPoint** can be specified at the **JourneyPattern** level, where it shall be used without further modification on all the **VehicleJourneys** that reference that **JourneyPattern**. Alternatively, it can be specified within the relevant **VehicleJourney**.

A **LayoverPoint** doesn't specify where in the trip the vehicle lays over. In order to provide timing information to real time systems, the layover point shall be used in an **EndDeadRun** (where the vehicle transits empty to the layover point at the end of a **VehicleJourney**) and a **StartDeadRun** (where the vehicle transits empty from the layover point to the start of the next **VehicleJourney**).

Table 28 - Permitted usage of elements within a LayoverPoint

Element Name	**Data Type**	**Used in TXC-PTI**
Duration	*DurationType*	Mandated if **LayoverPoint** is included
Name	*NaturalLanguageString*	Mandated if **LayoverPoint** is included
Location	*LocationStructure*	Mandated if **LayoverPoint** is included, indicating the coordinates of the layover point. For *LocationStructure* see section 7.4
MinimumDuration	*DurationType*	Optional

9.7.3 Dead runs

Dead runs in a **VehicleJourney** are similarly helpful for real time systems to estimate times for following trips. TXC provides both a **StartDeadRun** and an **EndDeadRun** for dead runs at the start and end of trips, and these are both allowable in TXC-PTI. Dead runs can take place from garage to the first stop for that vehicle, from the last stop for that vehicle to the garage, or between stops at the end of one trip and the start of another.

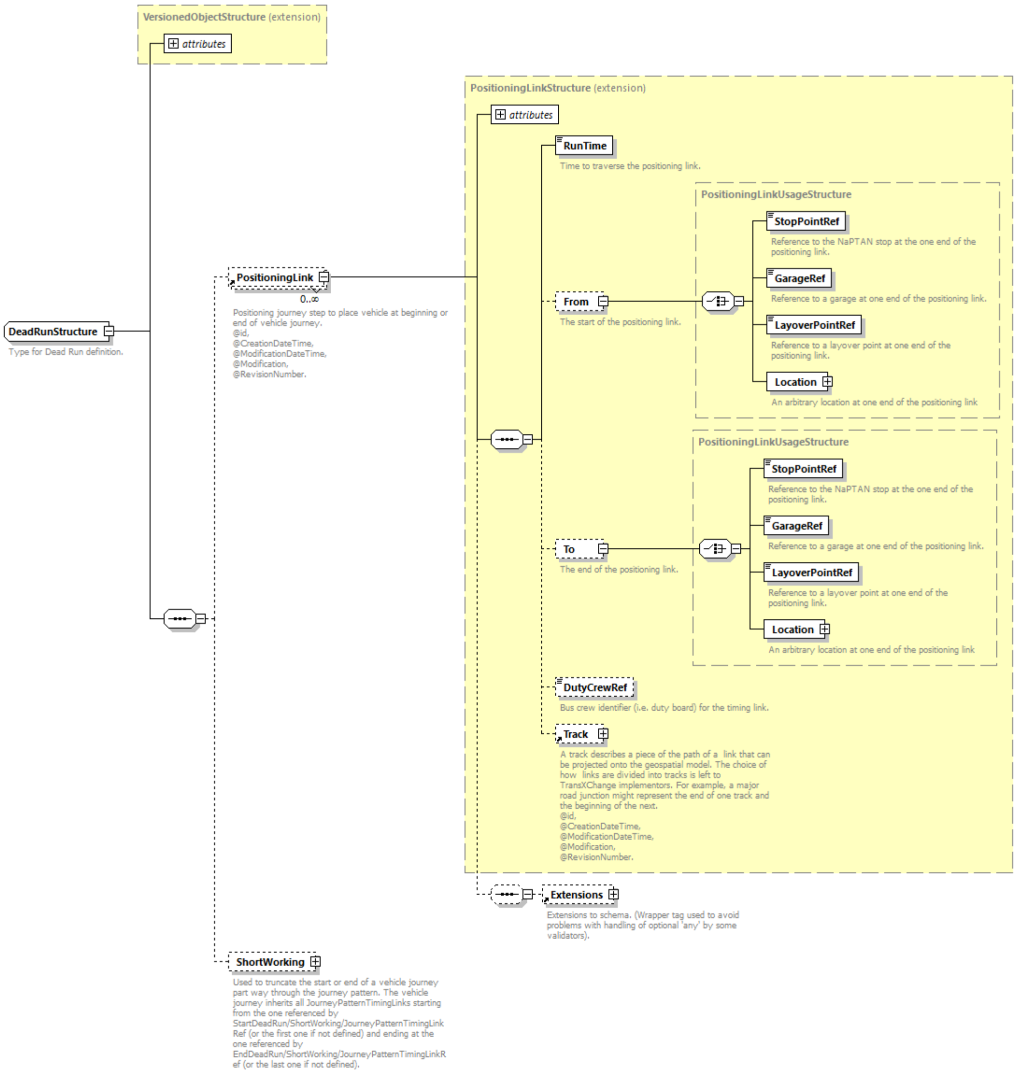

VersionedObjectStructure (extension)
attributes
DeadRunStructure
Type for Dead Run definition.
PositioningLink
0..∞
Positioning journey step to place vehicle at beginning or end of vehicle journey.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
PositioningLinkStructure (extension)
attributes
RunTime
Time to traverse the positioning link.
From
The start of the positioning link.
PositioningLinkUsageStructure
StopPointRef
Reference to the NaPTAN stop at the one end of the positioning link.
GarageRef
Reference to a garage at one end of the positioning link.
LayoverPointRef
Reference to a layover point at one end of the positioning link.
Location
An arbitrary location at one end of the positioning link
To
The end of the positioning link.
PositioningLinkUsageStructure
StopPointRef
Reference to the NaPTAN stop at the one end of the positioning link.
GarageRef
Reference to a garage at one end of the positioning link.
LayoverPointRef
Reference to a layover point at one end of the positioning link.
Location
An arbitrary location at one end of the positioning link
DutyCrewRef
Bus crew identifier (i.e. duty board) for the timing link.
Track
A track describes a piece of the path of a link that can be projected onto the geospatial model. The choice of how links are divided into tracks is left to TransXChange implementors. For example, a major road junction might represent the end of one track and the beginning of the next.
@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
ShortWorking
Used to truncate the start or end of a vehicle journey part way through the journey pattern. The vehicle journey inherits all JourneyPatternTimingLinks starting from the one referenced by StartDeadRun/ShortWorking/JourneyPatternTimingLinkRef (or the first one if not defined) and ending at the one referenced by EndDeadRun/ShortWorking/JourneyPatternTimingLinkRef (or the last one if not defined).

Figure 33 - The DeadRun structure

The structure of a dead run is shown in Figure 33 above. In TXC-PTI:

- there shall be a **PositioningLink**, which shall:
    - have a **RunTime** describing the run time between the **From** and **To** points;
    - have **From** and **To** elements, each of which shall be populated with either a **StopPointRef**, a **GarageRef**, a **LayoverPoint**, or an arbitrary **Location**, although this last is discouraged in TXC-PTI unless absolutely necessary.
    - As with route links, the use of **Track** element to describe the actual path that the vehicle takes is strongly recommended, but not mandatory in TXC-PTI.
- **ShortWorking** shall not be used. Instead, a **JourneyPattern** which fully defines the short working shall be defined.

Given a specific time in TXC, and with a structure comprised of link run times, a system will calculate a call time at any given point by adding all of the run times and wait times together and using the result as an offset from the departure time. There is no expectation that a system will have to subtract anything. On this basis, for the avoidance of doubt, TXC-PTI defines the **DepartureTime** of a **VehicleJourney** as either:

a) the departure time from the first stop on the **VehicleJourney**, where there is no **StartDeadRun**; or

b) the departure time of the **StartDeadRun**, where there is a **StartDeadRun**.

10 Flexible services

10.1 Declaring a flexible service

A **FlexibleService** is constructed, at a high level, in essentially the same way as a **StandardService**, and forms part of the **Service**, generally replacing the **StandardService**. In a **FlexibleService** there is a **FlexibleJourneyPattern** instead of a **JourneyPattern**. For this reason, a **FlexibleService** in TXC-PTI will have the same constraints as a **StandardService**, and a **FlexibleService** shall have at least one **FlexibleJourneyPattern**.

In TXC-PTI, a flexible service shall have a **FlexibleService** which shall consist of at least one **FlexibleJourneyPattern**.

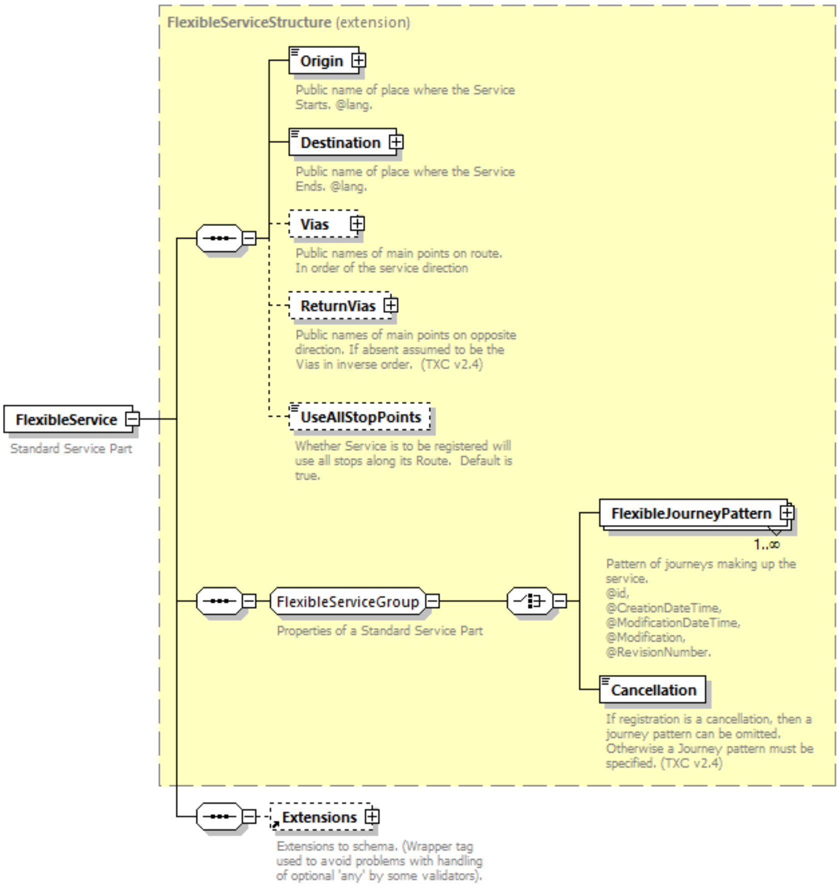

FlexibleServiceStructure (extension)
Origin
Public name of place where the Service Starts. @lang.
Destination
Public name of place where the Service Ends. @lang.
Vias
Public names of main points on route. In order of the service direction
ReturnVias
Public names of main points on opposite direction. If absent assumed to be the Vias in inverse order. (TXC v2.4)
FlexibleService
Standard Service Part
UseAllStopPoints
Whether Service is to be registered will use all stops along its Route. Default is true.
FlexibleJourneyPattern
1..∞
Pattern of journeys making up the service. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
FlexibleServiceGroup
Properties of a Standard Service Part
Cancellation
If registration is a cancellation, then a journey pattern can be omitted. Otherwise a Journey pattern must be specified. (TXC v2.4)
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 34 - The FlexibleService element

10.2 The FlexibleJourneyPattern element

At the top level, there is a high degree of commonality between a standard (non-flexible) service and a flexible service. The *CommonInterchangeGroup* is, in fact, common between both and a flexible service shall therefore follow the same rules for these elements – see section 8.2.

Beyond that, the flexible service is defined solely as an ordered set of stops, using the **StopPointsInSequence** element, and information shall be provided as to the booking arrangements needed to book the flexible service.

TXC-PTI does not need, at this time, to specify how these elements shall be completed. This will be the subject of a later revision.

In TXC-PTI, a **FlexibleJourneyPattern** shall consist of a set of stopping points, and shall provide information as to how the service shall be booked.

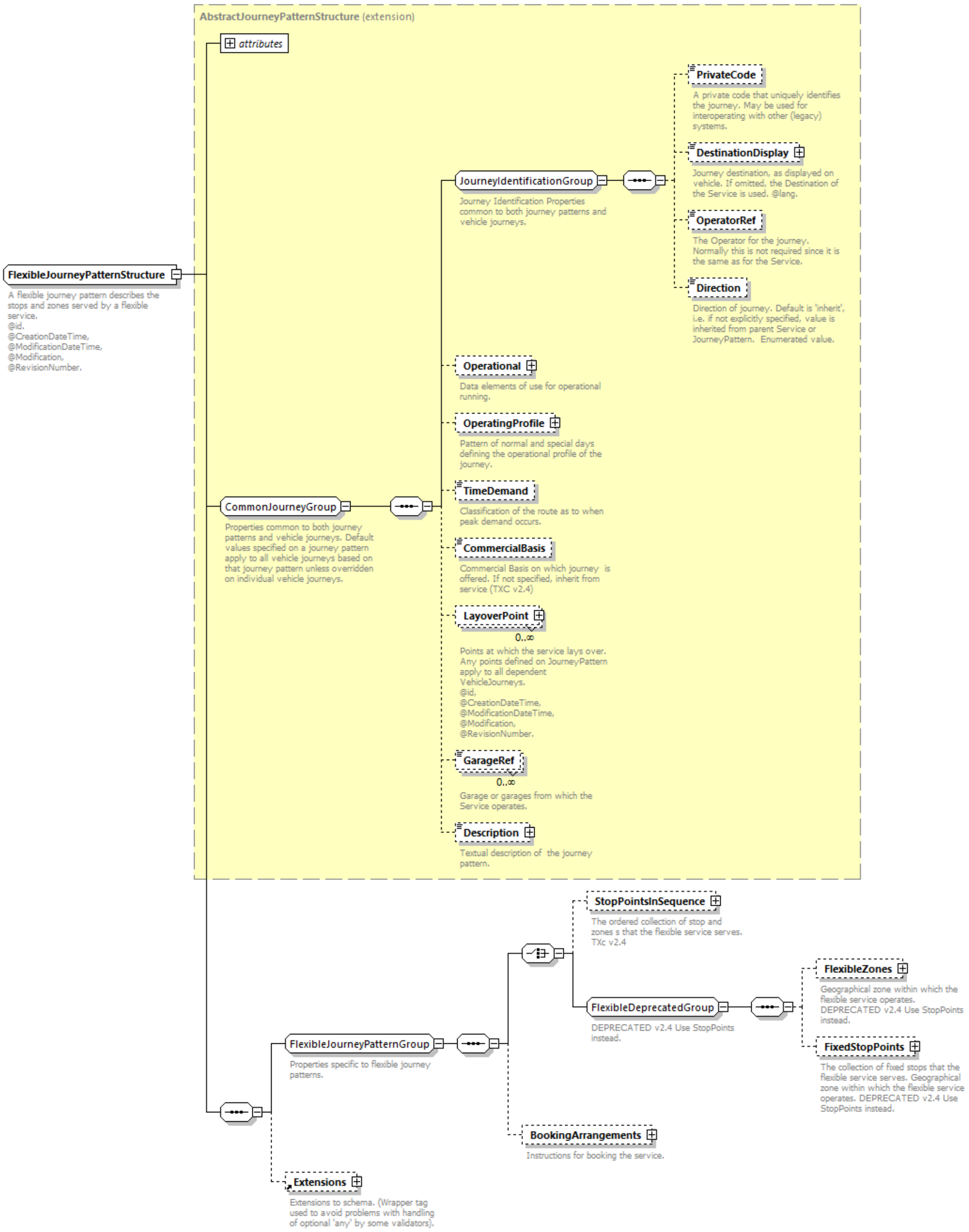

AbstractJourneyPatternStructure (extension)
attributes
FlexibleJourneyPatternStructure
A flexible journey pattern describes the stops and zones served by a flexible service. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
JourneyIdentificationGroup
Journey Identification Properties common to both journey patterns and vehicle journeys.
PrivateCode
A private code that uniquely identifies the journey. May be used for interoperating with other (legacy) systems.
DestinationDisplay
Journey destination, as displayed on vehicle. If omitted, the Destination of the Service is used. @lang.
OperatorRef
The Operator for the journey. Normally this is not required since it is the same as for the Service.
Direction
Direction of journey. Default is 'inherit', i.e. if not explicitly specified, value is inherited from parent Service or JourneyPattern. Enumerated value.
CommonJourneyGroup
Properties common to both journey patterns and vehicle journeys. Default values specified on a journey pattern apply to all vehicle journeys based on that journey pattern unless overridden on individual vehicle journeys.
Operational
Data elements of use for operational running.
OperatingProfile
Pattern of normal and special days defining the operational profile of the journey.
TimeDemand
Classification of the route as to when peak demand occurs.
CommercialBasis
Commercial Basis on which journey is offered. If not specified, inherit from service (TXC v2.4)
LayoverPoint
0..∞
Points at which the service lays over. Any points defined on JourneyPattern apply to all dependent VehicleJourneys. @id, @CreationDateTime, @ModificationDateTime, @Modification, @RevisionNumber.
GarageRef
0..∞
Garage or garages from which the Service operates.
Description
Textual description of the journey pattern.
StopPointsInSequence
The ordered collection of stop and zones s that the flexible service serves. TXc v2.4
FlexibleDeprecatedGroup
DEPRECATED v2.4 Use StopPoints instead.
FlexibleZones
Geographical zone within which the flexible service operates. DEPRECATED v2.4 Use StopPoints instead.
FixedStopPoints
The collection of fixed stops that the flexible service serves. Geographical zone within which the flexible service operates. DEPRECATED v2.4 Use StopPoints instead.
FlexibleJourneyPatternGroup
Properties specific to flexible journey patterns.
BookingArrangements
Instructions for booking the service.
Extensions
Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 35 - The flexible journey pattern structure