Department for Transport
TransXChange – An XML Standard for the Data Exchange of Bus Schedules and Related Information.

TransXChange Schema Guide

2.5
© Crown Copyright 2000-2014

The content in this document may be reproduced free of charge in any format or media without requiring specific permission, subject to the TransXChange Terms & Conditions of use, viewable at http://www.transxchange.org.uk. This is subject to the material not being used in a derogatory manner or in a misleading context. The source of the material must be acknowledged as Crown copyright and the title of the content must be included when being reproduced as part of another publication or service.
CONTENTS
Section

Preamble

CONTENTS

Section

Page

VERSION HISTORY

2

CONTENTS

3

LIST OF FIGURES

11

LIST OF TABLES

15

1 INTRODUCTION

19

1.1 Antecedents

19

1.2 Document Structure

20

1.3 Intellectual Property Rights

21

1.3.1 TransXChange Schema

21

1.3.1.1 TransXChange Schedules

21

1.3.1.2 TransXChange Document Publisher

21

1.4 Versioning

22

1.5 Naming Conventions

22

1.6 Presentation Conventions Used in the Schema Guide

22

1.6.1 XML Elements in Text

22

1.6.1.1 UML Diagrams

22

1.6.1.2 XML Structure Diagrams

23

1.6.1.3 Element Structure – Sequence

23

1.6.1.4 Element Structure – Choice

23

1.6.1.5 Multiplicity and Optionality

23

1.7 Summary of Changes

24

1.7.1 Changes in Release 2.5 of TransXChange

24

1.7.2 Changes in Release 2.4 of TransXChange

25

1.7.3 Changes in Release 2.2a of TransXChange.

26

1.7.4 Major Changes in Release 2.0 of TransXChange

26

1.8 Evolving TransXChange

26

1.9 Acknowledgments

27

1.10 Related Transport Information Standards

28

1.11 Legislation

28

1.12 Related Documents

28

2 OVERVIEW OF TRANSXCHANGE

30

2.1 The Purpose of TransXChange

30

2.2 TransXChange Components

30

2.3 Document Validation

31

2.4 How is TransXChange Used?

32

2.4.1 Registration of a Route with VOSA for EBSR

32

2.4.2 Update of a Registration with VOSA

32

2.4.3 General Purpose Exchange of Data

33

2.4.4 General Purpose Exchange of Data changes

33

2.5 Differences between the Schemas

34

3 SHORT TOUR OF THE TRANSXCHANGE ESSENTIAL MODEL

35

3.1 Representing a Bus Service in TransXChange

35

3.1.1 The NaPTAN Stop Model

36

3.1.1.1 The NaPTAN Stop Model Introduction

36
3.1.1.2 The NaPTAN Stop Model Details 38
3.1.2 Resolving NaPTAN Stop References 39
3.1.3 Variable Stop Allocations 39
3.1.4 Stop Types 40
3.1.5 NaPTAN Bus Stop Types 41
3.1.6 NPTG Administrative Support Types 42
3.1.7 NPTG Locality Support Types 42
3.1.8 NaPTAN Support Types 43
3.1.9 NaPTAN Location Support Types 45

3.2 The Route and Service Supply Model 46
3.2.1 Model Layer Concerns 48
3.2.2 Summary of Route & Supply Model Elements 49
3.2.3 Projection between Levels of Discourse 49

3.3 Route Model 50
3.3.1 Introduction to the Route Model 50
3.3.2 Route Model Details 51
3.3.3 Tracks 51
3.3.4 Track Example 52
3.3.5 Route Model Hierarchy 53

3.4 Journey Pattern Model 54
3.4.1 Introduction to the Journey Pattern Model 54
3.4.2 Journey Pattern Model Details 55
3.4.3 Journey Pattern Model Hierarchy 56

3.5 Vehicle Journey Model 57
3.5.1 Introduction to the Vehicle Journey Model 57
3.5.2 Vehicle Journey Model Details 59
3.5.3 Connecting Vehicle Journey Model 60
3.5.4 Vehicle Journey Model Hierarchy 61

3.6 The Use of Links in TransXChange 62
3.6.1 Structure Example of a Schedule with one Pattern and Two Journeys, 63
3.6.2 Structure Example of a Schedule with an Express Journey 64
3.6.3 Plotting a route on a Map 64

3.7 Inheriting Timing Link Values 65
3.7.1 Schedule and Journey Terms and Definitions 67
3.7.1.1 Time Related Terms 67
3.7.1.2 Routing Related Terms 67
3.7.2 Computation of Passing Times 68
3.7.2.1 Example of Inheritance of Passing Times 69
3.7.2.2 Rounding of Passing Times 70

3.8 Standard Services 71
3.8.1 Introduction to the Service Model 71
3.8.2 Introduction to the Standard Service Model 71
3.8.3 Standard Service Properties 73
3.8.4 Standard Service Components 74
3.8.5 Service Model Hierarchy 75
3.8.6 Service Model Types 76

3.9 Flexibly Routed Services 77
3.9.1 Introduction to the Flexible Service Model 77
3.9.2 Flexible Service Model Details 79
3.9.3 Flexibly Routed Service Hierarchy 79

3.10 Line Model 81
3.10.1 Introduction to the Line Model 81
3.10.2 Line Model Details 82
3.10.3 Line Hierarchy 82

3.11 Interchanges 83
3.11.1 Inheriting Interchange Values 84
3.11.2 Interchange Schematic 85
3.11.3 Interchange Instance Example 85
### Contents

**3.12 Fare Stages** 88  
**3.13 Dead Runs** 89  
3.13.1 Use of Dead Runs for Short Working 90  
3.13.2 Dead Run Hierarchy 90  
**3.14 The Registration Model** 91  
3.14.1 Introduction to the Registration Model 91  
3.14.2 Registration Model Details 92  
3.14.3 Registration Model Workflow States 94  
3.14.4 Registration Model Hierarchy 95  
3.14.5 Registration Model Support Types 96  
3.14.6 Populating a Registration 96  
3.14.7 Cancellation a Registration 96  
**3.15 Operators** 97  
3.15.1 Introduction to the Operator Model 97  
3.15.2 Operator Model Details 97  
3.15.3 Operator / Accessibility Booking ((v2.5) 99  
3.15.4 Operator Model Hierarchy 100  
3.15.5 Operator Model Support Types 101  
**3.16 Further Modelling Topics** 101  
3.16.1 Direction: Handling Inbound and Outbound Schedules. 101  
3.16.2 Modelling Complex Routes 103  
3.16.2.1 Services with Topologically Complex Routes 103  
3.16.2.2 Services with Complex Temporal Operational Patterns 106  
3.16.3 Modelling Services Efficiently 107  
3.16.3.1 Overall Reuse of Elements 108  
3.16.3.2 Inefficiencies in TransXChange 108  
3.16.3.3 Use of Sections 109  
3.16.4 Presenting Schedules in Timetables 110  
3.16.4.1 Using a Sequence Number 111  
3.16.4.2 Example of a Timetable using StopSequence 112  
3.16.5 Grouping Journeys within a Timetable 114  
3.16.5.1 Journey Grouping Model Overview 114  
3.16.5.2 JourneyGrouping Elements 115  
3.16.5.3 JourneyGrouping Model Hierarchy 115  
3.16.6 Specifying services that run before or after midnight 116  
**3.17 DayShift Examples** 117  
3.17.1 Late night Weekday service – Positive DayShift 117  
3.17.1.1 Actual Journeys 117  
3.17.1.2 As Shown in Beds with Positive DayShift 117  
3.17.2 Early morning Weekday Service – Negative DayShift 118  
3.17.2.1 Actual Journeys 118  
3.17.2.2 As shown in Beds with Positive DayShift 118  
3.17.3 Associating Operational Data with a Timetable 118  
3.17.3.1 Operational Data Model Introduction 119  
3.17.3.2 Operational Data Model Details 119  
3.17.4 Operational Model Hierarchy 120  
3.17.4.1 Operational Data Model Support Types 120  
3.17.4.2 Vehicle Equipment Model (+TXC v2.4) 120  
3.17.4.3 Assistance Service Model (+TXC v2.5) 122  
3.17.5 Service Descriptions 126  
**3.18 Modelling Operational Days** 126  
3.18.1 Specifying When the Service Operates – Summary 126  
3.18.2 Regular Operation – OperatingProfile 128  
3.18.3 Exceptional Operation – OperatingProfile 128  
3.18.4 Services that Run for Specific ServicedOrganisation Working Days 130  
3.18.4.1 ServicedOrganisation Model Introduction 130  
3.18.4.2 ServicedOrganisation Model Details 132  
3.18.5 General Principles for Using Operational Days 133
3.18.6 Footnotes 134
3.18.6.1 Example - Footnotes 134
3.18.6.2 Footnote Variations 134
3.18.6.3 Serviced Organisation Calendar Example 135
3.18.7 Frequent Services 138
3.18.8 Frequency Based Services 138
3.18.8.1 Frequency Described by Interval 138
3.18.8.2 Departure Described by Minutes Past Hour 139
3.18.8.3 Frequency Described on Multiple Individual Journeys – Merged Frequency 139
3.18.8.4 Multi-journey to single group, multiple frequencies 140
3.18.9 Partial Frequency Based Services (+TXC v2.4) 140
3.18.9.1 Text Descriptions for Frequency service 141
3.18.10 Service Operational Days 141
3.18.11 Structure Example of Schedule with Operational Day Exceptions 142

3.19 Calendar Model 142
3.19.1.1 Calendar Model Introduction 142
3.19.1.2 Calendar Model Details 143
3.19.2 Calendar Model Hierarchy 143

3.20 Summary of TransXChange Entities and Identifiers 145
3.20.1 Summary of TransXChange Entity Identifier Types 146
3.20.2 Summary of NPTG Entity Identifier Types 146
3.20.3 Private codes 146
3.20.4 Referencing Elements 147
3.20.5 Summary of TransXChange Ancillary Identifiers 147

3.21 Common Data types 148
3.21.1 XML Base Types 148
3.21.2 General Utility Types 148

3.22 Data Rights Model 149
3.22.1 Data Rights Model - Introduction 149
3.22.2 Data Rights Model - Introduction 150

4 WORKED EXAMPLE OF A TRANSXCHANGE SCHEDULE 151
4.1 Worked Example: Bus Timetable 151
4.2 Worked Example: Service Components 151
4.3 Worked Example: Operator 151
4.4 Worked Example: Registration 152
4.5 Worked Example: StopPoints 152
4.6 Worked Example: Route and Tracks 152
4.7 Worked Example: JourneyPattern 154
4.8 Worked Example: Line 155
4.9 Worked Example: VehicleJourney 155
4.10 Worked Example: Operational Times 155

5 EXAMPLES 156
5.1 Feature Examples 156
5.2 Real Examples 156

6 TRANSXCHANGE SCHEMA 158
6.1 TransXChange Schema Overview 159
6.2 TransXChange Root Element 160
6.2.1 TransXChange Element Attributes 161
6.2.2 TransXChange Root Attributes 161
6.2.3 TransXChange Child Elements 163
6.3 Stop & Topographical Elements – StopPoints and Zones 165
6.3.1 NptgLocalities Element 165
6.3.2 AnnotatedNptgLocalityRef Element 165
6.3.3 StopPoints Element 166
6.3.4 AnnotatedStopPointRef Element 166
6.3.5 StopPoint Element (Stop) 167
6.3.6 StopArea Element 167

6.4 **Network Elements – Routes and Tracks** 169

6.4.1 Route Element 169
6.4.2 RouteSection Element 169
6.4.3 RouteLink Element 170
6.4.4 Route / Track Element 172
6.4.5 Track Subelements 172
6.4.5.1 Track / Mapping Element 172
6.4.5.2 Track / Instructions Element 173
6.4.5.3 Track / Instructions / Feature Element 173

6.5 **Registration Elements: Operator, Registration, ShortNoticeRegistration** 175

6.5.1 Operators Element 175
6.5.2 Operator Element 175
6.5.3 LicensedOperator Element 177
6.5.4 Operator & LicensedOperator: Subelements 179
6.5.4.1 Operator / OperatorNamesGroup 179
6.5.4.2 Operator / OperatorLicenceGroup 179
6.5.4.3 Operator / OtherLicences Element (+TXC v2.4) 180
6.5.4.4 Operator / OperatorParentGroup (+TXC v2.4) 180
6.5.4.5 Operator / OperatorInfoGroup (+TXC v2.4) 181
6.5.4.6 Operator / RegionalOperatorRef (+TXC v2.4) 182
6.5.4.7 Operator / OperatorContactGroup 182
6.5.4.8 Operator / AccessibilityBooking / (+TXC v2.5) 183
6.5.4.9 Operator / Garages Element 185
6.5.4.10 Operator / ContributorGroup (+TXC v2.4) 185
6.5.4.11 DataRight / (+TXC v2.4) 186
6.5.5 Registration Element 187
6.5.6 Registration / RegistrationSubmissionGroup 188
6.5.7 Registration / RegistrationWorkflowGroup 189
6.5.8 Registration / RegistrationInfoGroup 190
6.5.9 Registration Subelements 191
6.5.9.1 Registration / VosaRegistrationNumber Element 191
6.5.9.2 Registration / SubmissionAuthor Element 192
6.5.9.3 Registration / TrafficArea Element 192
6.5.9.4 Registration / CirculatedAuthorities Element 193
6.5.9.5 Registration / SubsidyDetails Element 197
6.5.9.6 Registration / ContractedService Element 197
6.5.9.7 Registration / SupportingDocument Element 198
6.5.10 ShortNoticeRegistration Element 199
6.5.11 ShortNoticeRegistration / ChangeImpactGroup 199
6.5.12 ShortNoticeRegistration / ChangeJustificationGroup 199
6.5.13 ShortNoticeRegistration Subelements 200
6.5.13.1 ShortNoticeRegistration / Public Availability Element 200
6.5.13.2 ShortNoticeRegistration / ChangeImpact Element 201
6.5.13.3 ShortNoticeRegistration / ChangeToConnectAlteredService Element 201
6.5.13.4 ShortNoticeRegistration / ReplaceDiscontinuedService Element 201
6.5.13.5 ShortNoticeRegistration / LocalHolidayChange Element 202
6.5.13.6 ShortNoticeRegistration / SpecialOccasion Element 202
6.5.13.7 ShortNoticeRegistration / RegulationComplianceElement 202
6.5.13.8 ShortNoticeRegistration / ChangeRequestedByExternalAuthority Element 203
6.5.13.9 ShortNoticeRegistration / ExceptionalRequirement Element 203

6.6 **Service Description Elements** 204

6.6.1 Services Element 204
<table>
<thead>
<tr>
<th>6.6.2</th>
<th>Service Element</th>
<th>204</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6.3</td>
<td>Service / ServiceInfoGroup</td>
<td>206</td>
</tr>
<tr>
<td>6.6.4</td>
<td>Service / ServiceDescriptionGroup</td>
<td>207</td>
</tr>
<tr>
<td>6.6.5</td>
<td>Service / ServiceComponentGroup</td>
<td>208</td>
</tr>
<tr>
<td>6.6.6</td>
<td>Service / Subelements</td>
<td>209</td>
</tr>
<tr>
<td>6.6.6.1</td>
<td>Service / Line Element</td>
<td>209</td>
</tr>
<tr>
<td>6.6.6.2</td>
<td>Line / Description Element (+TXC v2.4)</td>
<td>210</td>
</tr>
<tr>
<td>6.6.6.3</td>
<td>LineColour Group (+TXC v2.4)</td>
<td>211</td>
</tr>
<tr>
<td>6.6.6.4</td>
<td>Service / OperatingPeriod Element</td>
<td>212</td>
</tr>
<tr>
<td>6.6.6.5</td>
<td>Service / ServiceClassification Element</td>
<td>212</td>
</tr>
<tr>
<td>6.6.6.6</td>
<td>Service / ServiceOperational Group</td>
<td>213</td>
</tr>
<tr>
<td>6.6.6.7</td>
<td>ServiceOperationalGroup / AssistanceService Element (+TXC v2.5)</td>
<td>214</td>
</tr>
<tr>
<td>6.6.6.8</td>
<td>ServiceOperationalGroup / ServiceFacilitySet Element (+TXC v2.5)</td>
<td>216</td>
</tr>
<tr>
<td>6.6.6.9</td>
<td>Service / AssociatedOperators Element</td>
<td>219</td>
</tr>
<tr>
<td>6.6.6.10</td>
<td>Service / StopRequirements Element</td>
<td>219</td>
</tr>
<tr>
<td>6.6.6.11</td>
<td>Service / ServiceAvailability Element</td>
<td>220</td>
</tr>
<tr>
<td>6.6.6.12</td>
<td>Service / ToBeMarketedWith Element</td>
<td>220</td>
</tr>
<tr>
<td>6.6.7</td>
<td>Service / JourneyGroupings (TXC v2.4)</td>
<td>221</td>
</tr>
<tr>
<td>6.6.7.1</td>
<td>Service / Built In Journey Grouping Element (TXC v2.4)</td>
<td>221</td>
</tr>
<tr>
<td>6.6.7.2</td>
<td>Service / CustomJourneyGrouping Element</td>
<td>222</td>
</tr>
<tr>
<td>6.7</td>
<td>StandardService, JourneyPattern, VehicleJourney</td>
<td>224</td>
</tr>
<tr>
<td>6.7.1</td>
<td>StandardService Element</td>
<td>224</td>
</tr>
<tr>
<td>6.7.2</td>
<td>StandardService / Subelements</td>
<td>225</td>
</tr>
<tr>
<td>6.7.2.1</td>
<td>StandardService / Vias Element</td>
<td>225</td>
</tr>
<tr>
<td>6.7.3</td>
<td>JourneyPatterns</td>
<td>226</td>
</tr>
<tr>
<td>6.7.4</td>
<td>JourneyPattern Element</td>
<td>226</td>
</tr>
<tr>
<td>6.7.4.1</td>
<td>JourneyPattern / CommonJourneyGroup</td>
<td>227</td>
</tr>
<tr>
<td>6.7.4.2</td>
<td>JourneyPattern / JourneyIdentificationGroup</td>
<td>228</td>
</tr>
<tr>
<td>6.7.5</td>
<td>JourneyPattern Subelements</td>
<td>229</td>
</tr>
<tr>
<td>6.7.5.1</td>
<td>JourneyPattern / StandardJourneyPatternGroup</td>
<td>229</td>
</tr>
<tr>
<td>6.7.5.2</td>
<td>CommonJourneyGroup / Operational / OperationalVariant Element</td>
<td>229</td>
</tr>
<tr>
<td>6.7.5.3</td>
<td>CommonJourneyGroup / Block Element</td>
<td>230</td>
</tr>
<tr>
<td>6.7.5.4</td>
<td>CommonJourneyGroup / VehicleType Element</td>
<td>230</td>
</tr>
<tr>
<td>6.7.5.5</td>
<td>VehicleType / Equipment Element (+TXC v2.5)</td>
<td>231</td>
</tr>
<tr>
<td>6.7.5.6</td>
<td>VehicleType / PassengerInfoEquipment Element (+TXC v2.4)</td>
<td>231</td>
</tr>
<tr>
<td>6.7.5.7</td>
<td>VehicleType / AccessVehicleEquipment Element (+TXC v2.4)</td>
<td>232</td>
</tr>
<tr>
<td>6.7.5.8</td>
<td>VehicleType / WheelchairEquipment Element (+TXC v2.4)</td>
<td>234</td>
</tr>
<tr>
<td>6.7.5.9</td>
<td>CommonJourneyGroup / Operational / TicketMachine Element</td>
<td>235</td>
</tr>
<tr>
<td>6.7.5.10</td>
<td>CommonJourneyGroup / LayoverPoint Element</td>
<td>236</td>
</tr>
<tr>
<td>6.7.6</td>
<td>JourneyPatternSection Element</td>
<td>236</td>
</tr>
<tr>
<td>6.7.7</td>
<td>JourneyPatternTimingLink Element</td>
<td>237</td>
</tr>
<tr>
<td>6.7.7.1</td>
<td>JourneyPatternTimingLink / CommonTimingLinkGroup</td>
<td>237</td>
</tr>
<tr>
<td>6.7.7.2</td>
<td>JourneyPatternTimingLink / JourneyPatternTimingLink</td>
<td>238</td>
</tr>
<tr>
<td>6.7.8</td>
<td>JourneyPatternStopUsageStructure</td>
<td>239</td>
</tr>
<tr>
<td>6.7.8.1</td>
<td>JourneyPatternStopUsage / JourneyPatternStopUsageGroup</td>
<td>240</td>
</tr>
<tr>
<td>6.7.8.2</td>
<td>JourneyPatternStopUsage / JourneyPatternStopUsageGroup</td>
<td>241</td>
</tr>
<tr>
<td>6.7.8.3</td>
<td>VariableStopAllocations Element</td>
<td>242</td>
</tr>
<tr>
<td>6.7.8.4</td>
<td>StopAccessibility Element (+TXC v2.5)</td>
<td>243</td>
</tr>
<tr>
<td>6.7.9</td>
<td>JourneyPatternInterchange Element</td>
<td>244</td>
</tr>
<tr>
<td>6.7.9.1</td>
<td>JourneyPatternInterchange / CommonInterchangeGroup</td>
<td>245</td>
</tr>
<tr>
<td>6.7.9.2</td>
<td>JourneyPatternInterchange / InterchangeInfoGroup</td>
<td>246</td>
</tr>
<tr>
<td>6.7.9.3</td>
<td>JourneyPatternInterchange / JourneyPatternInterchangeGroup</td>
<td>246</td>
</tr>
<tr>
<td>6.8</td>
<td>VehicleJourney Element</td>
<td>248</td>
</tr>
<tr>
<td>6.8.1</td>
<td>VehicleJourney / VehicleJourneyGroup</td>
<td>248</td>
</tr>
<tr>
<td>6.8.2</td>
<td>VehicleJourney / StandardVehicleJourneyGroup</td>
<td>249</td>
</tr>
<tr>
<td>6.8.3</td>
<td>Common VehicleJourney Subelements</td>
<td>250</td>
</tr>
<tr>
<td>6.8.3.1</td>
<td>VehicleJourney / DeadRun Element</td>
<td>250</td>
</tr>
<tr>
<td>6.8.3.2</td>
<td>VehicleJourney / PositioningLink Element</td>
<td>251</td>
</tr>
</tbody>
</table>
6.8.3.3 VehicleJourney / PositioningLink / PositioningStopUsageStructure 252
6.8.3.4 VehicleJourney / Frequency Element 253
6.8.3.5 Frequency Interval 253
6.8.4 VehicleJourneyTimingLink Element 254
6.8.4.1 VehicleJourneyTimingLink / VehicleJourneyTimingLinkGroup 254
6.8.5 VehicleJourneyTimingLink / VehicleJourneyStopUsage Element 255
6.8.6 VehicleJourney / VehicleJourneyInterchange Element 256
6.8.6.1 VehicleJourney / VehicleJourneyInterchangeGroup 257

6.9 FlexibleService, FlexibleJourneyPattern, FlexibleVehicleJourney 258
6.9.1 FlexibleService Element 258
6.9.1.1 FlexibleJourneyPattern Element 258
6.9.1.2 FlexibleJourneyPattern / FlexibleJourneyPatternGroup 259
6.9.2 FlexibleService Subelements 259
6.9.2.1 FlexibleService / FlexibleZones Element 259
6.9.2.2 FlexibleService / FixedStopPoints Element 260
6.9.2.3 FlexibleVehicleJourneyGroup / BookingArrangements Element 260
6.9.3 FlexibleVehicleJourney Element 261
6.9.3.1 FlexibleVehicleJourneyGroup / FlexibleServiceTimes Element 262

6.10 ConnectingVehicleJourney Element 263
6.10.1 VehicleJourney / ConnectingVehicleJourney Group 263
6.10.1.1 ConnectingVehicleJourney / ConnectingTimesGroup 264

6.11 OperatingProfile Element 265
6.11.1 Normal OperatingProfileGroup 265
6.11.2 Special OperatingProfileGroup 266
6.11.3 OperatingProfile Subelements 266
6.11.3.1 OperatingProfile / RegularDayType Element 266
6.11.3.2 OperatingProfile / RegularDayType / DaysOfWeek Element 266
6.11.3.3 OperatingProfile / PeriodicDayType / WeekOfMonth Element 267
6.11.3.4 SpecialDaysOperation Element: DaysOfOperation, DaysOfNonOperation 267
6.11.3.5 DateRange 268
6.11.3.6 OperatingProfile / BankHolidayOperation 268
6.11.3.7 OperatingProfile / BankHolidayElements 269
6.11.3.8 OperatingProfile / AllBankHolidayGroup 269
6.11.3.9 OperatingProfile / AllBankHolidayElement 269

6.12 ServicedOrganisation Element 271
6.12.1 ServicedOrganisation Subelements 273
6.12.1.1 ServicedOrganisation / DatePattern Element 273

6.13 Calendar Element (TXC v2.4) 274
6.13.1.1 Calendar / OperatingDay Element (TXC v2.4) 275
6.13.1.2 Calendar / DayType Element (TXC v2.4) 276

6.14 Miscellaneous Elements 276
6.14.1 SupportingDocument Element 276

7 COMMON SCHEMA ELEMENTS 278
7.1 LocationStructure 278
7.2 Duration Simple Type 278
7.3 TelephoneContactStructure Element 279
7.4 PostalAddressStructure Element 279
7.5 Note Element 281
7.6 Submode Element 282

8 ELECTRONIC BUS SERVICE REGISTRATION PROCESS 283
8.1 Step 1: Preparation 283
8.2 Step 2: Encoding 283
8.3 Step 3: Transmission 283
8.4 Step 4: Validation 283
### 8.5 Step 5: TAN Review

### 8.6 Step 6: Acceptance and Distribution

### 9 THE TRANSXCHANGE PUBLISHER

- **9.1 Required Environment**
- **9.2 Installation Process**
- **9.3 Run Time Options**
- **9.4 Generalised list of Publisher parameters**
- **9.5 Publishing Actions**
  - 9.5.1 Particulars & Service Matrix
  - 9.5.2 Route Map

### 10 NAMING & CODING CONVENTIONS

- **10.1 Naming of Elements**
  - 10.1.1 Use of Camel Case
  - 10.1.2 Use of Standard Name Suffixes
  - 10.1.3 Meaningful Names
  - 10.1.4 Semantically Significant Order
  - 10.1.5 Standardised Terminology
- **10.2 Typing of Elements**
- **10.3 Element Constraints**
- **10.4 Use of Attributes**
- **10.5 Implementation of Model Relationships**

### 11 NATIONAL LANGUAGE SUPPORT

- **11.1 Text Content Types**
  - 11.1.1 Use of Fixed Text
  - 11.1.2 Use of Free Text
  - 11.1.3 External Data
- **11.2 Publishing or Exchanging Documents**

### 12 VERSIONING

- **12.1 Version Numbering Convention**
- **12.2 Resource Versions**
  - 12.2.1 Schema URI Version
  - 12.2.2 Namespace URI Version
  - 12.2.3 Package Versions
- **12.3 Packages**
- **12.4 Version Identifiers & Change Tracking**
  - 12.4.1 Schema Version Identifier
  - 12.4.2 Indicating Versions on Data
  - 12.4.3 Data Element Version
  - 12.4.4 Change Trackable Entities
- **12.5 Exchange of Deltas**
- **12.6 Names of TransXChange Files**

### 13 TRANSMODEL & TRANSXCHANGE COMPARISON

- **13.1 Transmodel Principles**
- **13.2 Transmodel Terminology**
- **13.3 Divergences from Transmodel**
  - 13.3.1 TransXChange Representation of Journey Patterns
  - 13.3.2 Abbreviated Journey Patterns
  - 13.3.3 Groups of Links
14 INTEGRITY RULES .................................................................307
  14.1 Syntactic Integrity Rules ..................................................307
  14.2 Semantic Integrity Rules ..................................................309
  14.3 Ordered Relationships ....................................................312
  14.4 Precedence Rules for Combining General Date Elements ....313

15 APPENDIX A – REFERENCES TO OTHER STANDARDS ......315
  15.1 Transport Domain ..........................................................315
    15.1.1 NaPTAN & NPTG ......................................................315
    15.1.2 JourneyWeb ............................................................315
    15.1.3 Transmodel CEN TC 278 ............................................315
    15.1.4 Netex CEN TC 278 ....................................................315
  15.2 Software & General .......................................................315
    15.2.1 XML Schema ..........................................................315
    15.2.2 ISO Time Formats ....................................................315
    15.2.3 WGS 1984 Location Referencing ................................316
    15.2.4 ISO 639-1 Names of Languages ..................................316
    15.2.5 RFC 1766 Tags for the Identification of Languages ....316
    15.2.6 GovTalk XML Coding Standards ................................316
    15.2.7 UML Unified Modelling Language ...............................316

List of Figures
Figure 1-1 – XML Spy Diagram: Sequence ........................................23
Figure 1-2 – XML Spy Diagram: Choice ..............................................23
Figure 1-3 – XML Spy Diagram: Multiplicity ......................................24
Figure 2-1 – Overview of TransXChange Use ......................................32
Figure 2-2 – Common Set of Types in TransXChange Schemas ...........34
Figure 3-1 – UML Overview of TransXChange Model for a StandardService ....35
Figure 3-2 – UML Diagram of Elaboration of TransXChange model ..........36
Figure 3-3 – UML Diagram of Summary of Stop Model .....................37
Figure 3-4 – UML Diagram of selected NaPTAN Stop elements ..........38
Figure 3-5 – UML Diagram of Stop Classification Model ..................40
Figure 3-6 – UML Diagram of On-street bus Stop Classification Model ....41
Figure 3-7 – UML Diagram of NPTG Administrative Support types ..42
Figure 3-8 – UML Diagram of NPTG Locality Support types .............43
Figure 3-9 – UML Diagram of NaPTAN Support types ....................44
Figure 3-10 – UML Diagram of NaPTAN Location Support Types ....45
Figure 3-11 – UML Diagram of Route, JourneyPattern and VehicleJourney Models .............................47
Figure 3-12 – Service Model Layer Concerns ..................................48
Figure 3-13 – Correspondence between Links at Different Levels .......49
Figure 3-14 – UML Diagram of Route Model: Introduction ...............50
Figure 3-15 – UML Diagram of Route Model ..................................51
Figure 3-16 – UML Diagram of Route Model Element Hierarchy ......53
Figure 3-17 – UML Diagram of Journey Pattern Model: Introduction ...54
Figure 3-18 – UML Diagram of Journey Pattern Model: Details .......55
Figure 3-19 – UML Diagram of Journey Pattern Model Element Hierarchy ....56
Figure 3-20 – UML Diagram of Journey Pattern Model: Introduction ......58
Figure 3-21 – UML Diagram of Vehicle Journey Model: Details ........59
Figure 3-22 – UML Diagram of Connecting Vehicle Journey Model ....60
Figure 3-23 – UML Diagram of Vehicle Journey Model Element Hierarchy ......61
Figure 3-24 – Simple Route Map ..................................................63
Figure 3-25 – UML Diagram of Service Pattern elements ..................66
Figure 3-26 – Computation of Passing Times ....................................69
Figure 3-27 – UML Diagram of Service model: Introduction ..................71
Figure 3-28 – UML Diagram of Standard Service: Overview ............72
Figure 3-29 – UML Diagram of Standard Service: Details ...............73
Figure 3-30 – UML Diagram of Standard Service Parts .................74
Figure 3-31 – UML Diagram of Service Model Element Hierarchy ........................................... 75
Figure 3-32 – UML Diagram of Service Support types ............................................................. 76
Figure 3-33 – Flexible Network .................................................................................................. 77
Figure 3-34 – UML Diagram for Flexibly Routed Service: Introduction .................................. 78
Figure 3-35 – UML Diagram for Flexibly Routed Service: Details .......................................... 79
Figure 3-36 – UML Diagram of Flexibly Routed Service Element Hierarchy ......................... 80
Figure 3-37 – UML Diagram of Line Model ............................................................................... 82
Figure 3-39 – UML Diagram of Line Model Element Hierarchy ............................................. 82
Figure 3-38 – UML Diagram of Interchanges .......................................................................... 84
Figure 3-39 – Interchange Links .............................................................................................. 85
Figure 3-40 – UML Instance Diagram of Example Interchange .............................................. 87
Figure 3-41 – Fare Stages & Links ........................................................................................... 88
Figure 3-42 – UML Diagram of Dead Run Model ................................................................. 89
Figure 3-43 – UML Diagram of Dead Run Model Element Hierarchy .................................. 90
Figure 3-43 – UML Diagram of Basic Registration Model ...................................................... 91
Figure 3-44 – UML Diagram of TransXChange Registration .................................................. 93
Figure 3-45 – State transitions for registration Workflow Status ........................................... 94
Figure 3-46 – UML Diagram of Registration Element Hierarchy ........................................ 95
Figure 3-47 – UML Diagram of Registration Support types .................................................. 96
Figure 3-48 – UML Model of Operator .................................................................................. 97
Figure 3-49 – UML Diagram of TransXChange Operator Model .......................................... 98
Figure 3-50 – UML Diagram of TransXChange Operator Model .......................................... 99
Figure 3-51 – UML Diagram of Operator Element Hierarchy ............................................... 100
Figure 3-53 – Journey Directions ......................................................................................... 103
Figure 3-54 – Topology: Circular Route ................................................................................... 104
Figure 3-55 – Topology: Lollipop Route ................................................................................ 104
Figure 3-56 – Topology: Cloverleaf Route .............................................................................. 105
Figure 3-57 – Reuse of Elements ......................................................................................... 108
Figure 3-58 – Example of Sections ....................................................................................... 109
Figure 3-59 – Example: Use of Stop Sequencing .................................................................. 112
Figure 3-60 – UML Diagram of Service Journey Grouping elements - Introduction ............. 114
Figure 3-61 – UML Diagram of Service Journey Grouping elements – Details .................... 115
Figure 3-65 – UML Diagram of JourneyGrouping Element Hierarchy ............................... 116
Figure 3-62 – UML Diagram of Operational data elements: Introduction ............................ 119
Figure 3-63 – UML Diagram of Operational data elements: Details ..................................... 119
Figure 3-68 – UML Diagram of Operational Element Hierarchy ......................................... 120
Figure 3-65 – UML Diagram of VehicleEquipment Types .................................................... 121
Figure 3-66 – UML Diagram of Assistance Service Types ..................................................... 122
Figure 3-68 – UML Diagram of Normal Operation Profile .................................................... 128
Figure 3-69 – UML Diagram of Special Operation Profile ..................................................... 129
Figure 3-70 – UML Diagram of Serviced Organisation: Intro .............................................. 131
Figure 3-71 – UML Diagram of Serviced Organisation Days ................................................ 132
Figure 3-72 – Example of Footnotes ...................................................................................... 134
Figure 3-73 – Example LEA Calendar ................................................................................... 136
Figure 3-74 – Example School Calendar .............................................................................. 136
Figure 3-75 – UML Diagram of Calendar Model: Introduction ............................................ 143
Figure 3-76 – UML Diagram of Calendar Model: Details ..................................................... 143
Figure 3-82 – UML Diagram of Calendar Element Hierarchy ............................................. 144
Figure 3-77 – UML Diagram of XML base Data types ........................................................... 148
Figure 3-78 – Additional utility types used by TransXchange & NaPTAN ......................... 149
Figure 3-79 – UML Diagram of Rights Model: Introduction .............................................. 149
Figure 3-80 – UML Diagram of Rights Model: Details ......................................................... 150
Figure 4-1 – Worked Example: Map of the Route ................................................................. 153
Figure 4-2 – Worked Example: Journey Pattern ................................................................. 154
Figure 5-1 – Table List of TransXchange real examples ......................................................... 158
Figure 6-1 – UML Diagram of TransXChange Schema: Core elements ............................ 159
Figure 6-2 – UML Diagram of TransXChange Schema Overview of elements ................... 160
Figure 6-3 – Attributes for TransXChange Schema ............................................................... 161
Figure 6-4 – TransXChange Change management attributes .......................................................... 162
Figure 6-5 – Top Level Elements of TransXChange ........................................................................ 164
Figure 6-6 – NptgLocalities Element ......................................................................................... 165
Figure 6-7 – AnnotatedNptgLocalityRef Element ..................................................................... 166
Figure 6-8 – StopPoints Element ........................................................................................... 166
Figure 6-10 – Route Element .................................................................................................... 169
Figure 6-11 – RouteSection Element ........................................................................................ 170
Figure 6-12 – RouteLink Element ............................................................................................ 171
Figure 6-13 – Track Element ...................................................................................................... 172
Figure 6-14 – Mapping Element ............................................................................................... 173
Figure 6-15 – Instructions Element .......................................................................................... 173
Figure 6-16 – Operators Element ............................................................................................. 175
Figure 6-17 – Operators Element ............................................................................................. 175
Figure 6-18 – Operator Element .............................................................................................. 177
Figure 6-19 – LicensedOperator Element ................................................................................. 178
Figure 6-20 – OperatorNamesGroup ....................................................................................... 179
Figure 6-21 – Operator / OperatorLicenceGroup ..................................................................... 180
Figure 6-22 – Operator / OperatorLicence Element .................................................................. 180
Figure 6-23 – Operator / OperatorParentGroup ....................................................................... 181
Figure 6-24 – Operator / OperatorInfoGroup .......................................................................... 182
Figure 6-25 – Operator / RegionalOperatorRef .............................................................. 182
Figure 6-26 – Operator / OperatorContactGroup ..................................................................... 183
Figure 6-27 – Operator / AccessibilityBooking element ........................................................ 184
Figure 6-28 – Operator / Garages / Garage Element .............................................................. 185
Figure 6-29 – Operator / ContributorGroup Element .......................................................... 186
Figure 6-30 – Operator / DataRight Element ......................................................................... 187
Figure 6-32 – RegistrationSubmissionGroup ....................................................................... 189
Figure 6-33 – Registration Workflow Group .......................................................................... 190
Figure 6-35 – Registration / VosaRegistrationNumber Element ....................................... 192
Figure 6-36 – Registration / SubmissionAuthor Element ..................................................... 192
Figure 6-37 – Registration / TrafficArea Element ................................................................. 193
Figure 6-38 – Registration / CirculatedAuthorities Element ............................................ 197
Figure 6-39 – Registration / SubsidyDetails Element ............................................................ 197
Figure 6-40 – Registration / ContractedService Element .................................................... 198
Figure 6-41 – Registration / SupportingDocument Element .................................................. 198
Figure 6-42 – ShortNoticeRegistration Element ..................................................................... 199
Figure 6-43 – ShortNoticeRegistration / ChangeImpactGroup ............................................. 199
Figure 6-44 – ShortNoticeRegistration / ChangeJustificationGroup ....................................... 200
Figure 6-45 – ShortNoticeRegistration / PublicAvailabilityElement .................................... 201
Figure 6-46 – ShortNoticeRegistration / ChangeImpactElement ........................................... 201
Figure 6-47 – ShortNoticeRegistration / ChangeToConnectAlteredService Element .......... 201
Figure 6-48 – ShortNoticeRegistration / ReplaceDiscontinuedServiceElement .................. 202
Figure 6-49 – ShortNoticeRegistration / LocalHolidayChangeElement ............................... 202
Figure 6-50 – ShortNoticeRegistration / SpecialOccasionElement .......................................... 202
Figure 6-51 – ShortNoticeRegistration / RegulationOrderComplianceElement .................... 202
Figure 6-52 – ShortNoticeRegistration / ChangeRequestedByExternalAuthorityElement ...... 203
Figure 6-53 – ShortNoticeRegistration / ExceptionalRequirementElement ................................ 203
Figure 6-54 – Service Element ............................................................................................... 205
Figure 6-55 – Service / ServiceInfoGroup ............................................................................... 207
Figure 6-56 – Service / ServiceDescriptionGroup ............................................................... 208
Figure 6-57 – Service / ServiceComponentGroup ................................................................. 209
Figure 6-58 – Service / Line Element ..................................................................................... 210
Figure 6-59 – Service / Line Element ..................................................................................... 211
Figure 6-60 – Line / LineColourGroup ..................................................................................... 211
Figure 6-61 – Service / OperatingPeriodElement .................................................................... 212
Figure 6-62 – Service / ServiceClassificationElement .......................................................... 213
Figure 6-63 – Service / ServiceOperationalGroup .................................................................. 214
Figure 6-86 – AssistanceService Element (+TXC v2.5) ......................................................... 216
Figure 6-87 – ServiceFacilitySet Element (+TXC v2.5) .......................................................... 219
Figure 6-65 – Service / StopRequirements Element .............................................................. 220
Figure 6-66 – Service / ServiceAvailability Element .............................................................. 220
Figure 6-67 – Service / ToBeMarketedWith Element ............................................................ 221
Figure 6-68 – Service / JourneyGroupings Element ............................................................... 221
Figure 6-69 – Service / Built in Journey Grouping Element .................................................. 222
Figure 6-70 – Service / CustomJourneyGrouping Element .................................................... 223
Figure 6-71 – StandardService Element ................................................................................ 224
Figure 6-72 – Service / Vias Element ...................................................................................... 225
Figure 6-73 – JourneyPattern Element .................................................................................. 226
Figure 6-74 – JourneyPattern / CommonJourneyGroup ........................................................ 228
Figure 6-75 – JourneyIdentificationGroup ............................................................................ 229
Figure 6-76 – JourneyPattern / StandardJourneyPatternGroup ........................................... 229
Figure 6-78 – Operational / Block Element ........................................................................... 230
Figure 6-79 – Operational / VehicleType Element ................................................................. 231
Figure 6-80 – Operational / Abstract Equipment Element ..................................................... 231
Figure 6-82 – VehicleType / AccessVehicleEquipment Element (+TXC v2.4) .................. 234
Figure 6-83 – VehicleType / WheelchairVehicleEquipment Element (+TXC v2.4) .............. 235
Figure 6-85 – JourneyPattern / LayoverPoint Element .......................................................... 236
Figure 6-89 – JourneyPatternTimingLink Element ............................................................... 237
Figure 6-90 – JourneyPatternTimingLink / CommonTimingLinkGroup ............................... 238
Figure 6-91 – JourneyPatternTimingLink / JourneyPatternTimingLinkGroup ...................... 239
Figure 6-92 – JourneyPattern / JourneyPatternStopUsageStructure .................................. 240
Figure 6-93 – JourneyPattern / JourneyStopUsageGroup ..................................................... 241
Figure 6-94 – JourneyPattern / JourneyPatternStopUsageGroup ....................................... 242
Figure 6-95 – JourneyPattern / VariableStopAllocation Element ....................................... 242
Figure 6-96 – StopAccessibility Element .............................................................................. 244
Figure 6-97 – JourneyPatternInterchange Element ............................................................. 245
Figure 6-98 – CommonInterchangeGroup ........................................................................... 246
Figure 6-99 – JourneyPatternInterchange / InterchangeInfoGroup ...................................... 246
Figure 6-100 – JourneyPatternInterchange / JourneyPatternInterchangeGroup .................. 247
Figure 6-101 – VehicleJourney Element .............................................................................. 248
Figure 6-102 – VehicleJourney / VehicleJourneyGroup ....................................................... 249
Figure 6-103 – VehicleJourney / StandardVehicleJourneyGroup ......................................... 250
Figure 6-104 – VehicleJourney / DeadRun Element .............................................................. 251
Figure 6-105 – DeadRun / PositioningLink Element .............................................................. 252
Figure 6-106 – DeadRun / PositioningLinkUsageStructure .................................................. 253
Figure 6-107 – VehicleJourney / Frequency Element ............................................................ 253
Figure 6-108 – FrequencyInterval Element .......................................................................... 254
Figure 6-109 – VehicleJourneyTimingLinkElement .............................................................. 254
Figure 6-110 – VehicleJourneyTimingLinkGroup ................................................................. 255
Figure 6-111 – VehicleJourneyStopUsageElement ............................................................... 256
Figure 6-112 – VehicleJourneyInterchangeElement .............................................................. 257
Figure 6-113 – VehicleJourneyInterchangeGroup ................................................................. 257
Figure 6-114 – FlexibleServiceElement ............................................................................... 258
Figure 6-115 – FlexibleJourneyPatternElement .................................................................... 259
Figure 6-116 – FlexibleJourneyPatternElement .................................................................... 259
Figure 6-117 – FlexibleZones Element .................................................................................. 260
Figure 6-118 – FlexibleService / FixedStopPointsElement .................................................. 260
Figure 6-119 – FlexibleVehicleJourney / BookingArrangementsElement .............................. 261
Figure 6-120 – FlexibleVehicleJourney ............................................................................... 261
Figure 6-121 – FlexibleVehicleJourney / FlexibleServiceTimesElement ............................. 262
Figure 6-122 – ConnectingVehicleJourneyElement ............................................................... 263
Figure 6-123 – ConnectingVehicleJourneyElement ............................................................... 264
Figure 6-124 – ConnectingVehicleJourney / ConnectingTimesGroup ................................. 265
Figure 6-125 – OperatingProfileElement ............................................................................. 266
Figure 6-126 – OperatingProfile / RegularDayTypeElement ............................................ 266
Figure 6-127 – OperatingProfile / DaysOfWeekElement ....................................................... 267
# List of Tables

Table 1-1 – Forms for Registering Bus Services in England and Scotland ........................................ 29
Table 2-1 – Differences between Schemas .................................................................................................. 34
Table 3-1 – Resolving Stop References ........................................................................................................ 39
Table 3-2 – Correspondence between Links and Nodes ................................................................................. 49
Table 3-3 – Example Track Instructions ........................................................................................................ 53
Table 3-4 – Structure Example of a Schedule ................................................................................................. 63
Table 3-5 – Structure Example of Schedule: Shared Journey Pattern .......................................................... 64
Table 3-6 – Structure Example of Schedule: Express VehicleJourney ......................................................... 64
Table 3-7 – Journey Properties and Defaults ................................................................................................. 66
Table 3-8 – Example of Computation of Inherited Passing Times ................................................................. 70
Table 3-9 – Example of Rounding of Passing Times ....................................................................................... 70
Table 3-10 – Interchange Properties and Defaults ......................................................................................... 85
Table 3-11 – Example: Eye Timetable with Explicit Stop Sequencing ....................................................... 113
Table 3-12 – Example: Eye Timetable with Implicit Stop Sequencing ....................................................... 113
Table 3-13 – Positive DayShift example ....................................................................................................... 117
Table 3-14 – Positive DayShift example - effect .............................................................................................. 117
Table 3-15 – Negative DayShift example: ...................................................................................................... 118
Table 3-16 – Positive DayShift example: Monday to Friday Bed .................................................................. 118
Table 3-17 – Precedence of Entity Levels ..................................................................................................... 133
Table 3-18 – Precedence of Normal Operation Day Types ............................................................................ 134
Table 3-19 – Footnotes generated for different inputs .................................................................................. 135
Table 3-20 – Example Serviced Organisation Particulars for LEA ............................................................... 136
Table 3-21 – Example Serviced Organisation Particulars for School .......................................................... 137
Table 3-22 – Frequency Service Timetable representation: Interval .............................................................. 138
Table 3-23 – Example Frequent Service Timetable representation: Interval ................................................. 138
Table 3-24 – Example Frequent Service Timetable presentation: Minutes .................................................. 139
Table 3-25 – Multi-journey Representation of Frequency Based journeys ................................................. 139
Table 3-26 – Merged presentation of separate Frequency journeys having the same frequencies and end time ........................................................................................................................................... 140
Table 3-27 – Multi-journey Representation of Two Frequencies ................................................................. 140
Table 3-28 – Merged presentation of separate Journeys with different frequencies ..................................... 140
Table 3-29 – Representation of Partial Frequency Based journeys .............................................................. 140
Table 3-30 – Presentation of Partial Frequency Based journeys ................................................................. 140
Table 3-31 – Frequency service Text Descriptions ....................................................................................... 141
<table>
<thead>
<tr>
<th>Preamble</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3-32 – Main Entities of the TransXChange Model</td>
<td>146</td>
</tr>
<tr>
<td>Table 3-33 – TransXChange Main Entity Identifier Types</td>
<td>146</td>
</tr>
<tr>
<td>Table 3-34 – NPTG Main Entity Identifier Types</td>
<td>146</td>
</tr>
<tr>
<td>Table 3-35 – References to Entities in the TransXChange Model</td>
<td>147</td>
</tr>
<tr>
<td>Table 3-36 – Externally referenced Entities of the TransXChange Model</td>
<td>148</td>
</tr>
<tr>
<td>Table 4-1 – Worked Example: Bus Timetable</td>
<td>151</td>
</tr>
<tr>
<td>Table 4-2 – Worked Example: StopPoint Instances</td>
<td>152</td>
</tr>
<tr>
<td>Table 4-3 – Worked Example: RouteLink Instances</td>
<td>153</td>
</tr>
<tr>
<td>Table 4-4 – Worked Example: Timing Links for Journey Pattern</td>
<td>155</td>
</tr>
<tr>
<td>Table 4-5 – Worked Example: Timing Links for VehicleJourney 1A</td>
<td>155</td>
</tr>
<tr>
<td>Table 5-1 – TransXChange Examples</td>
<td>158</td>
</tr>
<tr>
<td>Table 6-1 – TransXChange Root element</td>
<td>160</td>
</tr>
<tr>
<td>Table 6-2 – Allowed Values for Link / Direction</td>
<td>170</td>
</tr>
<tr>
<td>Table 6-3 – Allowed Values for FeatureType</td>
<td>174</td>
</tr>
<tr>
<td>Table 6-4 – Allowed Values for RelativeBearing</td>
<td>174</td>
</tr>
<tr>
<td>Table 6-5 – Allowed Values for LicenceClassification</td>
<td>179</td>
</tr>
<tr>
<td>Table 6-6 – Allowed Values for LicenceStatus</td>
<td>180</td>
</tr>
<tr>
<td>Table 6-7 – Allowed Values for Data Owner</td>
<td>182</td>
</tr>
<tr>
<td>Table 6-8 – Allowed Values for AssistanceAvailability</td>
<td>183</td>
</tr>
<tr>
<td>Table 6-9 – Allowed Values for BookingMethod</td>
<td>184</td>
</tr>
<tr>
<td>Table 6-10 – Allowed Values for Policy Status</td>
<td>185</td>
</tr>
<tr>
<td>Table 6-11 – Allowed Values for Contributor Classification</td>
<td>186</td>
</tr>
<tr>
<td>Table 6-12 – Allowed Values for AllowedUse</td>
<td>186</td>
</tr>
<tr>
<td>Table 6-13 – Allowed Values for PolicyStatus</td>
<td>186</td>
</tr>
<tr>
<td>Table 6-14 – Allowed Values for Registration / ApplicationClassification</td>
<td>189</td>
</tr>
<tr>
<td>Table 6-15 – Allowed Values for Registration / RegistrationWorkflowStatus</td>
<td>190</td>
</tr>
<tr>
<td>Table 6-16 – Allowed Values for Registration / RegistrationVariantNature</td>
<td>190</td>
</tr>
<tr>
<td>Table 6-17 – Allowed Values for TanCode</td>
<td>191</td>
</tr>
<tr>
<td>Table 6-18 – Allowed Values for TrafficArea / Names</td>
<td>193</td>
</tr>
<tr>
<td>Table 6-19 – Allowed Values for CirculatedAuthorityNames</td>
<td>197</td>
</tr>
<tr>
<td>Table 6-20 – Allowed Values for SubsidyType</td>
<td>197</td>
</tr>
<tr>
<td>Table 6-21 – Allowed Values for Service / Mode</td>
<td>206</td>
</tr>
<tr>
<td>Table 6-22 – Allowed Values for Service / CommercialBasis</td>
<td>206</td>
</tr>
<tr>
<td>Table 6-23 – Allowed Values for Service / Direction</td>
<td>208</td>
</tr>
<tr>
<td>Table 6-24 – Allowed ServiceClassification Combinations</td>
<td>213</td>
</tr>
<tr>
<td>Table 6-25 – Allowed Values for AssistanceServiceType (+TXC v2.5)</td>
<td>214</td>
</tr>
<tr>
<td>Table 6-26 – Allowed Values for Staffing (+TXC v2.5)</td>
<td>214</td>
</tr>
<tr>
<td>Table 6-27 – Allowed Values for AccessibilityTools (+TXC v2.5)</td>
<td>215</td>
</tr>
<tr>
<td>Table 6-28 – Allowed Values for EmergencyServices (+TXC v2.5)</td>
<td>215</td>
</tr>
<tr>
<td>Table 6-29 – Allowed Values for CateringFacilityList</td>
<td>217</td>
</tr>
<tr>
<td>Table 6-30 – Allowed values for ServiceFacilitySet / FareClasses (+TXC v2.4)</td>
<td>217</td>
</tr>
<tr>
<td>Table 6-31 – Allowed values for ServiceFacilitySet / NuisanceFacility (+TXC v2.5)</td>
<td>217</td>
</tr>
<tr>
<td>Table 6-32 – Allowed values for ServiceFacilitySet / PassengerCommsFacilityList (+TXC v2.5)</td>
<td>217</td>
</tr>
<tr>
<td>Table 6-33 – Allowed values for ServiceFacilitySet / PassengerInformationFacility (+TXC v2.5)</td>
<td>217</td>
</tr>
<tr>
<td>Table 6-34 – Allowed values for ServiceFacilitySet / SafetyFacility (+TXC v2.5)</td>
<td>218</td>
</tr>
<tr>
<td>Table 6-35 – Allowed values for ServiceFacilitySet / SanitaryFacility (+TXC v2.5)</td>
<td>218</td>
</tr>
<tr>
<td>Table 6-36 – Allowed values for ServiceFacilitySet / AccommodationFacility (+TXC v2.5)</td>
<td>218</td>
</tr>
<tr>
<td>Table 6-37 – Allowed values for ServiceFacilitySet / LuggageCarriage (+TXC v2.5)</td>
<td>218</td>
</tr>
<tr>
<td>Table 6-38 – Allowed values for ServiceFacilitySet / Reservations (+TXC v2.4)</td>
<td>219</td>
</tr>
<tr>
<td>Table 6-39 – Allowed Values for TimeDemand</td>
<td>227</td>
</tr>
<tr>
<td>Table 6-40 – Allowed Values for Service / CommercialBasis</td>
<td>227</td>
</tr>
<tr>
<td>Table 6-41 – Allowed Values for JourneyPattern / Direction</td>
<td>229</td>
</tr>
<tr>
<td>Table 6-42 – Allowed Values for PassengerInfoEquipment / PassengerInfo</td>
<td>232</td>
</tr>
<tr>
<td>Table 6-43 – Allowed Values for MobilityNeed</td>
<td>233</td>
</tr>
<tr>
<td>Table 6-44 – Allowed Values for AssistanceNeeded</td>
<td>233</td>
</tr>
<tr>
<td>Table 6-45 – Allowed Values for AssistedBoardingLocation</td>
<td>233</td>
</tr>
</tbody>
</table>
Table 6-46 – Allowed Values for TimingLink / CommercialBasis .......................... 238
Table 6-47 – Allowed Values for VehicleJourney / Direction ............................. 239
Table 6-48 – Allowed Values for Activity ....................................................... 240
Table 6-49 – Allowed Values for TimingStatus ............................................... 241
Table 6-50 – Allowed Values for LimitationStatus ......................................... 243
Table 6-51 – Accessibility defaults by mode ................................................. 243
Table 6-52 – Allowed Values for TransferMode ............................................. 245
Table 6-53 – Allowed Values for InterchangeActivity ..................................... 247
Table 6-54 – AllBankHolidays by Country ..................................................... 270
Table 6-55 – Allowed values for Serviced Organisation Type ......................... 272
Table 6-56 – Allowed values for DateClassification ....................................... 273
Table 7-1 – Allowed characters for Address Line ......................................... 281
Table 10-1 – TransXChange Attributes ......................................................... 291
Table 11-1 – Elements That May Contain Natural Language Text ................. 293
Table 12-1 – TransXChange 2.4 Module Names .......................................... 298
Table 12-2 – TransXChange Document Version Attributes ........................... 299
Table 12-3 – Entity Change Tracking Attributes .......................................... 300
Table 12-4 – TransXChange Tracked Data Elements .................................... 301
Table 13-1 – Comparison of Key Transmodel Terms ..................................... 304
Table 14-1 – Syntactic Integrity Rules .............................................................. 309
Table 14-2 – Severity Codes for Semantic Integrity Rules ............................. 309
Table 14-3 – Intrinsic & Extrinsic Semantic Integrity Rules ......................... 312
Table 14-4 – Ordered Relationships ............................................................... 313
Table 14-5 – Date Elements in Order of Precedence .................................... 314
1  INTRODUCTION

TransXChange is a UK national data standard for the exchange of bus route and timetable information. The standard is sponsored by the UK Department for Transport and is mandated by the Vehicle Operating Services Agency (VOSA) for the electronic registration of UK bus services with Traffic Area Offices (TAO) and Local Authorities.

TransXChange allows the exchange of route and timetable information between, amongst others:

- Bus Service Operators
- Traffic Area Offices
- Local Authorities
- Passenger Transport Executives
- Traveline – the National Passenger Transport Information System
- Suppliers of AVL (Automatic Vehicle Location) and delivery systems

TransXChange comprises a set of W3C XML schemas with related documentation and other tools.

This Schema Guide is intended to provide a technical overview and reference manual to TransXChange for system developers, data providers and other users of TransXChange.

The Guide is accompanied by a set of worked examples, available at the www.transxchange.org.uk web site. These provide explanations, diagrams and XML for using every feature of TransXChange. A summary table of the examples is given in Section 5.

Note that detailed documentation of individual schema elements is provided as annotations within the schema itself. Software Tools such as XML SPY can be used to explore the structure and details of the schema.

1.1  Antecedents

Version 1.0 of TransXChange was originally developed by Cap Gemini in 2001 for the Traffic Area Network (TAN) under contract to the UK Department for Transport. The TransXChange model for public transport schedules was based on Transmodel, the European standard reference Data Model for Public Transport. Transmodel is intended:

- To promote a common integrated approach in the design of public transport information systems.
- To provide an open architecture for such systems.
- To provide a general model that can easily be adapted to create specific implementations.
- To support the reliable exchange of information between different software products.

An early version of Transmodel provided a starting point for the TransXChange Logical Reference Model that underpins the TransXChange XML schema. As a comprehensive, supplier-neutral, general purpose information model for transport information, Transmodel provides a valuable overall context of concepts and terminology extending over most aspects of public transport information (see Section 13). However, it should be noted that Transmodel is an abstract model, and it covers a wider scope of function than that required for TransXChange. Furthermore, Transmodel was expressed primarily in terms of an Entity-Relationship model, without the benefits of the encapsulation and richer constraints available in an Object-based language such as XML. A concrete XML implementation such as TransXChange must make a specific interpretation of the subset of Transmodel that is salient for its objectives, and must use the data types and other capabilities of its technology. The main divergences from Transmodel terminology are listed in section 13.2.

Subsequent updates, also managed by Cap Gemini, developed revised releases 1.1, 1.2, & 1.2.1
TransXChange version 2.0, (2003-2004) was a major revision of the standard, managed by Carl Bro and Kizoom, which included harmonisation with government standards for XML schemas, and addressed a number of issues arising from early-adopters' experience of the initial version.

2.1 (2005) was a very minor update to 2.0 to harmonise with other changes to NaPTAN. All TransXChange 2.1 documents should be fully compatible with 2.1 tools for import.

2.2 (end 2008) was a minor update to 2.1 to enable the publishing of route maps and to demonstrate multi-level version support for the publisher.

2.4 (2010) is an extensive update with a number of new features as described later below. It includes extensive revisions to this document including a new set of UML diagrams. It has been remodularised internally to facility maintenance and further alignment with Transmodel. An updated version of the publisher is available.

2.5 (2013) is an minor update to 2.4 with a number of new features to support in particular accessibility as described later below. It includes revisions to this document including an updated set of UML diagrams, recoloured to align with NeTEx’s use of colours to indicate different functional groups of conceptual elements. An updated version of the publisher is required.

The TransXChange Publisher, a tool used to produce human readable timetables from TransXChange documents was provided with release 2.0 and onwards. A new enhanced version of the publisher was produced in 2008 including a desktop interface. This includes support for a multiple schema levels: the latest includes including 2.1, 2.2a and 2.4 versions of the schema.

1.2 Document Structure

The TransXChange Concept Guide is organised as follows:

Part I – Introduction & Overview.
The chapters in Part I are intended to give a summary of the basic concepts and purpose of TransXChange:

- Information about the TransXChange Concept Guide.
- The Purpose of TransXChange.
- TransXChange Basic Concepts.
- TransXChange Logical Model.

Part II – Worked Examples.
Part II provides an example of the components of a TransXChange document.

- Simple Worked Example.
- It also provides an index to the systematic set of examples demonstrating the use of all TransXChange features that may be found at the web site.

Part III – Schema Structure.
The chapters in Part III provide a detailed account of the TransXChange Schema elements:

- Topographical Elements: Stops & Localities.
- Network Elements: Routes & Tracks.
- Service Description Elements.
- Operational Date & Time Elements.

The chapters in Part IV provide technical details on various aspects of TransXChange documents and technology:

- Technical Annexes.
  - Registration Process
  - TransXChange Publisher.
1.3 Intellectual Property Rights

1.3.1 TransXChange Schema

*TransXChange* is Crown Copyright, managed by the UK Department for Transport. The schema may be used without charge.

The *TransXChange* Schema may reference other Schemas that are also Crown Copyright, or that are owned by Associate Members of the UK Government GovTalk initiative.

Anyone who wishes to reproduce the Schema in any format must acknowledge the source and state that the Schema are the copyright of the named Associate Member or Crown Copyright, as appropriate. The permission to reproduce does not extend to any Schema or parts of Schema which are specifically identified as being the copyright of anyone who is not a Member or Associate Member. Permission to reproduce these Schema or parts of these Schemas must be obtained from the identified copyright holders.

*TransXChange* is based on open source software standards, notably XML.

The designated owner of the *TransXChange* schema for GovTalk is:

TransXChange, Transport Direct
Department for Transport,
2/17 Great Minster House,
33 Horseferry Road,
London
SW1P 4DR

1.3.1.1 TransXChange Schedules

Rights in the contents of bus schedules encoded as *TransXChange* conformant XML documents are separate from rights in the *TransXChange* Schema itself. Document content is the property of the publisher of each document.

Data usage rights may be specified within a document for individual elements or for the whole document (+TXC 2.4). *TransXChange* 2.4 introduces tags for recording the data rights in a document. See later.

1.3.1.2 TransXChange Document Publisher

*TransXChange* includes a software tool, the *TransXChange Publisher*, which may be used to transform XML schedules into PDF output. *TransXChange Publisher*, is supplied on a free-to-use licence on an unwarranted, ‘as-is’ basis. The publisher runs under a Java environment (JRE 1.4.2 or higher).

The Publisher may use on-line web services to fetch stop and map data that are incorporated into the published output. The use of stop data and map data in published output is governed by the terms of use of the publisher. In particular the map data may only be used for validating *TransXChange* documents submitted to the EBSR process, and not for other commercial uses such as publicity material, planning, etc.
1.4 Versioning

A strict versioning system is used for TransXChange, following e-Gif principles. This was made explicit in Version 2.0 of TransXChange and is explained in Section 12.1.

1.5 Naming Conventions

Systematic Naming conventions are used for schema elements following the e-Gif guidelines. The conventions are described in Section 8.

1.6 Presentation Conventions Used in the Schema Guide

1.6.1 XML Elements in Text

TransXChange uses the XML Schema Language (See http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/ and http://www.w3.org/TR/xmlschema-2/) and its terminology, such as “sequence” and “choice” to formally describe its data structures.

Throughout this TransXChange Schema Guide:

- XML elements are shown in bold italic type, for example the JourneyPattern element.
- XML attributes are shown in bold, for example MappingSystem.
- Containment of a subelement by another element is shown by a forward slash, for example StopPoint / AtcoCode.

1.6.1.1 UML Diagrams

Unified Modelling Language (UML) notation is used for class and instance diagrams to show the formal structure of the TransXChange Logical Reference model; the diagrams express structure in terms of classes, connected by association, aggregation and inheritance relationships, corresponding to the semantics available in XML’s built-in reference and extension mechanisms. Note that the UML diagrams are provided for explanatory purposes only, and omit an amount of detail (in particular, only a few element properties are typically shown as class attributes, and intermediary elements of a relationship are sometimes omitted.). UML notation uses well known conventions for showing the navigability, multiplicity, etc., of model elements, which we do not repeat here.

Note that in UML structure diagrams we label relationships in the direction of the navigability. Most relationships are navigable in only one direction, indicated by the arrow that points in the direction of navigability, i.e. coming from the entity that holds reference, to the referenced entity.

For TransXChange, we refine the standard UML conventions by the systematic use of colour: in particular:

- Network topology elements are shown in diagrams in green (for example, Route, StopPoint).
- Service level and service pattern related elements are shown in yellow (for example, FlexibleService, JourneyPattern, JourneyPatternTimingLink).
- Vehicle journey related elements are shown in orange (for example, VehicleJourney, VehicleJourneyTimingLink).
- Elements concerned with operational days, dates and times are shown in blue, (for example, OperatingProfile, BankHolidays, Frequency).

Different levels of detail are shown in the UML diagrams; introductory diagrams omit details and provide a high level overview; model diagrams show detailed attributes including physical attributes used to implement relationships; hierarchical views show the supertypes of objects; supporting diagrams show the low level data types used in the model diagrams.

Since we are depicting a physical model, in detailed diagrams we also indicate the attributes used to implement relationships.
1.6.1.2 XML Structure Diagrams

XML Spy (from Altova GmbH) structure diagrams are used extensively in the detailed schema description to illustrate the containment structure of XML schema fragments. Each XML element is shown as a solid box. Use of a complex data type is shown by a dashed box. The presence of attributes is indicated by a ‘+’. Since a common set of metadata attributes is used for first class objects, we do not generally show the attributes, though they may be listed in the accompanying documentation, using a convention of including the attribute name in the element comment prefixed by an ‘at’ sign (‘@’), for example ‘@lang’.

1.6.1.3 Element Structure – Sequence

The hexagonal symbol with the horizontal line of three dots indicates “sequence of.” For example, Figure 1-1 says the element ValidityPeriod consists of the sequence of StartTime followed by EndTime. Both elements are defined in the namespace whose prefix is “txc”. The adornment of a small series of horizontal lines in their upper left box corners indicates that StartTime and EndTime have a simple type. Types are normally shown in the bottom half of the box.

![Figure 1-1 – XML Spy Diagram: Sequence](image)

1.6.1.4 Element Structure – Choice

The hexagonal symbol with the switch-like icon indicates a choice. For example in Figure 1-2 there is a choice between the elements NoSubsidy and Subsidy. Subsidy has a further substructure, indicated by a “+” in at the right-hand end. NoSubsidy is simple type.

![Figure 1-2 – XML Spy Diagram: Choice](image)

1.6.1.5 Multiplicity and Optionality

Whether elements are required or optional, and the multiplicity (cardinality) of elements is indicated by adornments as follows:

- A *line dashed line* on the connecting line and surrounding box indicates an element is optional. For example, in Figure 1-3: FlexibleZones and Description.
- A *solid line* indicates a mandatory element. For example, in Figure 1-3: StopPointRef.
- A *number adornment* indicates a multiplicity other than one. ‘Many’ is indicated by an infinity sign ∞. Thus, for example in Figure 1-3, there may be zero or one Activity instances per StopUsage, but there can be between one and many StopUsages per FlexibleZone.
1.7 Summary of Changes

1.7.1 Changes in Release 2.5 of TransXChange

TransXChange 2.5 includes a small number of enhancements to improve support for accessibility and to integrate NaPTAN 2.5 changes to support Eire. In order to provide an underlying conceptual model and facilitate a long term harmonisation, the changes are based on NeTEx elements wherever possible.

- Functional
  - PTIC-083 NaPTAN change: Support for Eire locations on NaPTAN stops:
    - ITM (Irish Transverse Mercator) allowed as grid type.
    - Multiple Grid translations allowed.
  - PTIC-086 NaPTAN change: StopAccessibility added to StopPoint to allow stop level overrides of accessibility data.
  - PTIC-087 Accessibility Support
    - IFP 156 – Operator Booking Lines:
      - Operator extended with AccessibilityBooking information element.
    - IF145 Mode Accessibility:
      - Add new elements to WheelchairEquipment, booking required, guide dog, etc.
      - Add AssistanceService to Operational group and to Service.
      - Add StopAccessibility to StopUsage. Allows stop level overrides
      - Add Availability Condition to Equipment.
  - Align with JourneyWeb:
    - Allow a Transport Submode to be specified.
    - ServiceFacilitySet added to Operational data describe service properties such as fare classes, catering, toilets, aligned with JourneyWeb. uSes a subset fo NeTEX.

- Documentation
  - Emphasize that journeys with the same operateor, line and journey pattern should be in the same service.
  - Emphasize that VehicleJourney values common to other journeys should be placed on the JourneyPattern and not on individual journeys.
  - Give guidance on populating StartDate and EndDate for Service OperatingPeriods.

Figure 1-3 – XML Spy Diagram: Multiplicity
Emphasize that **Special Days** should only be used for immediate exceptions.
- Emphasize that **RunTime** values should be given for a **JourneyPatternTimingLinks** in order to enable precise journey times.
- Emphasize that **DeadRuns** with **RunTime** values should be provided in order to facilitate accurate real-time calculation of service times.

**Technical**
- The version number attribute on a TransXChange document was previously a fixed value (e.g. 2.1, 2.4, etc.). It is now a variable that defaults to the current value (e.g. 2.5). This makes it easier for implementers to use a single schema binding with documents that conform to earlier releases.

### 1.7.2 Changes in Release 2.4 of TransXChange

**TransXChange** includes a number of semantic revisions to add in features requested by the PTIC TransXChange user group and to enable further harmonisation.

**Functional**
- PTIC-XXXX-Authority Changes Addition of ExeterCity & NorwichCity Areas, modularisation of Areas to a separate schema file.
- PTIC-001 Add Additional National Operator Database attributes to Operator.
- PTIC-002 Add Partial frequent Service Interval.
- PTIC-003 Relax constraints on service classification.
- PTIC-011 Temporal grouping of post-midnight journeys.
- PTIC-012 Explicit Journey Grouping.
- PTIC-013 Line Description by Direction
- PTIC-016 Additional business rule validation.
- PTIC-018 Support concise cancellation.
- PTIC-022 Footnote publishing & Serviced Organisations.
- PTIC-027 Multiple Operational References per Journey.
- PTIC-028 Add workflow attributes. *
- PTIC-029 Vehicle Accessibility info.
- PTIC-031 Permission Levels & IPR Use *
- PTIC-032 Support Dynamic Vias on StopUsage.
- PTIC-033 Add Recommended End date to Service Operating Period.
- PTIC-035 Support for general school holidays
- PTIC-036 Support Minimum layover time on Layovers.
- PTIC-038 Add CommercialBasis flags to Service, etc.
- PTIC039 Improve support for JourneyInterchanges.
- PTIC-040 Support for Jan2ndDisplacementHoliday & StAndrewsDay.
- PTIC-041 Support for Line Colours.
- PTIC-042 Add Marketing Name to Service Description.
- PTIC-067 Permit use of & in Service Codes.
- PTIC-071 National Term database Support. +
- PTIC-074 Extensible Authority names

**Technical**
- T-001 Modularisation of TransXChange_common into discrete modules for each TransXChange component (JourneyPattern, operator etc.) to facilitate.
- T-002 Revision of all UML diagrams to use unified EA diagrams.
- T-003 Revision of all XML diagrams to show element types.
- T-004 Addition of Package modularisation diagrams.
- T-005 XML Refactoring of all ordinary objects as VersionedObject descendants. Addition of substitution groups to show hierarchy.
Elements that are added in v2.4 are marked (+TXC 2.4). Elements that are changed are marked ‘Changed in v2.4’.

1.7.3 Changes in Release 2.2a of TransXChange.

TransXChange 2.2a included minor semantic revisions to enable the support of route maps.
- Addition of location to annotated Stop Refs.

1.7.4 Major Changes in Release 2.0 of TransXChange

TransXChange 2.0 included major syntactic and semantic revisions to bring it closer to NaPTAN and other standards. The following is a summary of major changes in release 2.0. See Section Error! Reference source not found. for a full list of changes.

- Modularisation.
- eGif GovTalk compliance.
- Data Integrity improved.
- Welsh Language support added.
- Route Links remodeled.
- VehicleJourney & JourneyPattern model revised for efficiency and integrity.
- Days of Operation standardised and extended.
- Registration Number supported.
- Provision of a full TransXChange Schema Guide with examples.
- New TransXChange Publisher to transform XML documents to Acrobat pdf format.
- Use of revised NaPTAN & NPTG models.
- Revision of Registration / Service relationship to enable connecting services to be specified in registrations.

New function for:
- New National Operator code, when available.
- Flexibly Routed Services.
- Vehicle Operations.
- School Dates.
- Fare Stages (but not fares).
- Dead Run support.
- Dynamic Bay Allocation.
- Add further descriptive elements to Service.

For changes in 2.1 see Appendix B.

Note that an extension of TransXChange to handle fares information, currently referred as FareXChange, is being considered for future development.

1.8 Evolving TransXChange

Successive versions of TransXChange introduce new features that are not present in previous versions: and documents containing these features may not be processed by tools that are designed for a previous version. However a principle of full upwards compatibility of data is upheld – the existing content used to create TransXChange documents should be exactly mappable to the revised schema at a new release. Since existing TransXChange documents are generated automatically by various suppliers’ tools, the enhancement of the tools to generate the new format should provide a straightforward upward migration path.

2.0 put into place a formal versioning method that allows concurrent operation of schemas at different levels.

Normally strict compatibility is achieved in each TransXChange release, that is, if the new feature is not used in the document, the document may be processed as if it was of an earlier version.
1.9 Acknowledgments

The first version document was been prepared as part of the TransXChange 2.0 release and updated for 2.1 by the Carlbro (Richard Mejia, Paul Robinson) and Kizoom teams (Nick Knowles, Tom White) under direction of Roger Slevin of the Department for Transport, and Tim Hughes (VOSA). Introduction, modelling, structure example, schema and technical sections have been provided by Kizoom, worked examples by Carlbro. We thank Matt Francis of Action Information Management Ltd for his examples, comments and suggestions including the table of comparative terminology. Thanks also to Andrew Cudbertson (Arriva), Ross Dixon (CGEY), Michael Forbes (Opcom), Kieran Holmes (Cap Gemini), Paul Houghton (Trandata), Peter Miller (ACIS), Peter Neil (Trapeze), Mike Ness (WSAtkins), Pete Ridley (Thales), John Prince (SYPTE), John Gallagher (Thales), Stephen Corlett (Thales), Richard Shaw (WSAtkins), Alex Worrel (AtkinsGlobal), Adrian Walters (Infocell), Mary Doonan (Journey Plan), Dave Walter (Anite), Dr Martin Siczkowski (WYPTE), Mike James (Tandata), John Pryer (Omnibus), Wilfred Düx (MDV), Graham Browne (WYPTE), Peter Stoner, and other ATCO, RTIG and PTIC members for their comments, examples and other feedback.

The 2.4 version document was been prepared by a Kizoom team (Nick Knowles, Chris Anderson) for Centaur (Mark Cartwright) under direction of Chris Gibbard & Roger Slevin of the Department for Transport. The PTIC user group has contributed the ideas and priorities for the new features over the past several years. The enhanced National Operator Code Model draws on papers by Mark Fell and others.
1.10 Related Transport Information Standards

TransXChange is an XML based standard and is compatible with the following standards for public transport information:

- **ATCO-CIF**: ATCO-CIF is a general purpose exchange format for common elements of timetable information. TransXChange is the successor to ATCO-CIF.

- **NaPTAN**: The National Public Transport Access Nodes database is a UK nationwide system for uniquely identifying all the points of access to public transport in the UK. The NaPTAN database is maintained centrally under contract to the Department for Transport. The NaPTAN standard is described in a separate document (see bibliography at end). NaPTAN is intended to assign every UK train station, coach terminus, airport, ferry terminal, bus stop, etc, a unique NaPTAN identifier. For large interchanges & termini, NaPTAN points identify the entrances from the public thoroughfare – one identifier is distinguished as the main entrance.

- **NPTG**: The National Public Transport Gazetteer is an auxiliary database to NaPTAN that provides a means of relating NaPTAN stops to UK towns and villages, as well as to the regional groupings used to manage Public Transport data. TransXChange assumes knowledge of the current NPTG database by all parties.

- **Transmodel**: Transmodel is an abstract Reference Data model of the data of interest to organisations designing transport related information systems. It has been developed through several European Commission sponsored projects. An XML version is currently under development as NeTEx (Network Exchange). Changes to TransXChange are designed to conform to NeTEx data structures.

- **JourneyWeb**: JourneyWeb is an XML protocol allowing distributed journey planning. The protocol is a UK national de facto standard sponsored by the UK Department for Transport, and is being used in the Transport Direct Portal project to provide contiguous distributed journey planning across the whole of the UK.

- **SIRI**: Service Interface for Real-time Information is a standard for the exchange of real time bus information between systems developed by CEN members of the UK Real Time Interest Group. It is also based on NaPTAN and Transmodel, and will be evolved so as to harmonise with other related standards including TransXChange.

- **UK Geocoding** References: For geospatial location references TransXChange supports both Grid references – using Eastings and Northings, with support for both UK Mainland and Irish grids – and WGS 84 Latitude and Longitude. However Grid location references must be used for registrations.

1.11 Legislation


1.12 Related Documents

A TransXChange Registration provides an electronic representation of the following forms issued by the Vehicle and Operating Services Agency (VOSA). The forms may be downloaded in pdf format from http://www.vosa.gov.uk/.

<table>
<thead>
<tr>
<th>Description</th>
<th>England and Wales</th>
<th>Scotland</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application to Register a Bus Service</td>
<td>PSV350</td>
<td>PSV350 (Scotland)</td>
<td>June 2003</td>
</tr>
<tr>
<td>Short Notice Registration Supplementary Form</td>
<td>PSV350A</td>
<td>PSV350A (Scotland)</td>
<td>Sept 2001</td>
</tr>
<tr>
<td>Local Bus Service Registration. Guide for Operators</td>
<td>PSV353A</td>
<td></td>
<td>June 2004</td>
</tr>
<tr>
<td>Application to Change or Cancel details of a Local Service Registration</td>
<td>PSV355</td>
<td>PSV355A (Scotland)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1-1 – Forms for Registering Bus Services in England and Scotland
2 OVERVIEW OF TRANSXCHANGE

2.1 The Purpose of TransXChange

TransXChange is a standard format for describing bus routes and schedules as XML documents that can be automatically exported and imported between different computer systems. The documents themselves can be exchanged by different transport mechanisms, for example, FTP, email or http, and can be zipped (compressed) significantly to speed transfer.

There are two main variants of TransXChange:

- **Registration schema**: Defines an XML document specifically for the purpose of registering bus services with VOSA. Each document contains a single registered "service". Includes integrity constraints to ensure the document elements are complete and consistent.
  - [http://www.transxchange.org.uk/schema/2.5/TransXchange_registration.xsd](http://www.transxchange.org.uk/schema/2.5/TransXchange_registration.xsd).

- **General schema**: Defines an XML document for exchanging bus timetables and related information for many different purposes. More than one bus service can be specified in a single document. Includes integrity constraints to ensure the document elements are complete and consistent. May be used to exchange particular elements by themselves.
  - [http://www.transxchange.org.uk/schema/2.5/TransXchange_general.xsd](http://www.transxchange.org.uk/schema/2.5/TransXchange_general.xsd).

In addition there is a third variant that can be used just for exchanging just the updates to a previously exchanged document:

- **Delta schema**: Defines an XML document for exchanging just the changes to a previously exchanged bus timetables or related information. The schema does not have integrity constraints to ensure the document elements are complete and consistent.
  - [http://www.transxchange.org.uk/schema/2.5/TransXchange_genera_delta.xsd](http://www.transxchange.org.uk/schema/2.5/TransXchange_genera_delta.xsd).

2.2 TransXChange Components

TransXChange comprises the following components:

- **TransXChange Schema**: A model and formal XML schema (and variants) for describing and encoding bus schedules as XML documents. The schema can be used with software tools to check that documents are correctly formatted and contain the required content.

- **TransXChange Documents and Process**: A description and explanation of the standard, including rules for creating, managing and using TransXChange documents with software tools.

- **TransXChange Publisher**: The publisher is a free tool issued along with the TransXChange, schemas which allows users to render TransXChange XML documents into a readable timetable-like layout, using an Acrobat pdf or html output file format. The free Acrobat reader from Adobe can be used to read and print pdf files. TransXChange Publisher requires the installation of a standard open source environment for running Java and XSLT – this can also be downloaded free. Use of these tools is described in Chapter 9. The TransXChange Publisher can be run in two modes: for Registration, in which case a specific subset of content is published for the registered particulars of a service and for General Use, which includes some additional content.

It should be emphasised that TransXChange is a data definition standard, and not a software program or a dynamic protocol in itself. It is intended to enable different suppliers and user communities to build systems that can share information correctly, cheaply and efficiently, but does not prescribe detailed error handling or other implementation details – such as the exact representational model - that will vary according to the requirements of individual applications.
Different applications may use different proprietary internal representations to store timetables exchanged with TransXChange.

### 2.3 Document Validation

To be valid *TransXChange* data, documents must satisfy two levels of validity criteria:

1. **Well-formedness and validity:** Documents must parse and validate against the *TransXChange* schema at the specified level – **Registration** or **General** – including all the integrity constraints coded within the schema, such as for key uniqueness and reference. Any document that does not satisfy the syntactic rules will be rejected for Registration and is very unlikely to be accepted or understood correctly in uses under the General Schema.

2. **Correctness:** Documents must satisfy additional processing rules and constraints that are not enforceable in the XML of the schema, but which are specified in this document, or as annotations in the schema (In case of any inconsistency, the schema should be regarded as definitive). Typically these rules cover additional complex processing or uniqueness constraints that cannot readily be expressed using XML’s built-in mechanisms. Any document that is not correct may be rejected for Registration and may not be accepted or understood correctly in uses on the General Schema. A number of semantic rules are listed later, and a severity assigned to them. The publisher provides a diagnostic function to checks for a number of these errors.
2.4 How is TransXChange Used?

The following three scenarios give the most common uses for TransXChange:

(i) To register a complete service.
(ii) To update a registration.
(iii) To exchange service related data for a wide variety of other purposes.

2.4.1 Registration of a Route with VOSA for EBSR

The most common scenario for use of TransXChange is to make a registration (Figure 2-1), with VOSA under the Electronic Bus Schedule Registration system and runs as follows:

1. Bus schedule data is prepared using scheduling software, including stop data from NaPTAN and route and geospatial data from other sources.
2. The schedule is exported as a TransXChange XML document to VOSA for registration. On export, the document is validated against a specified version of the schema. Note that TransXChange documents can also in principle be created by hand, though this would be both tedious and error prone.
3. The schedule is then imported by VOSA and Local Transport Authorities. On import, the document is validated against the version of the schema indicated by the document.
4. Following validation, the registered particulars alone are rendered as a readable pdf document using the Registration option of the TransXChange publisher.
5. The schedule is then imported by information system builders such as journey planners and AVL system implementers.
6. All or part of routes and schedules may be exchanged by system providers, annotated with additional operational data, over and above the registered particulars.

2.4.2 Update of a Registration with VOSA

TransXChange will also be commonly used to update an existing registration.

1. The schedule is updated by the owner using the schedule preparation system.
2. The schedule is reported as an XML registration document with updated data and modified change dates. Note that the whole schedule must be recreated; TransXChange does not.
Department for Transport
TransXChange Schema Guide

Part I
Introduction & Overview

Currently formally support the exchange of ‘deltas’, that is, changes to just part of a route or timetable (though this is likely to be added in future).

3. The schedule is revalidated and imported by VOSA, and the changed parts are updated in the VOSA database. The validation and propagation process thereafter is as for registration.

2.4.3 General Purpose Exchange of Data

TransXChange can also be used for the general purpose exchange of structured bus schedule data between any two information systems. Normally the TransXChange General schema will be used for this purpose, as it allows consistent subsets of data to be exchanged. Example uses might include:

- Exchanging schedule information with journey planning systems that wish to use the service.
- Exchanging route information with mapping systems that wish to draw the route.
- Exchanging schedule and operational data with AVL systems that wish to provide real-time bus predictions.
- Exchanging school term dates with Educational Authorities.
- Exchanging Operator details.

The precise scenario of use will depend on each specific purpose, but may be described generally (Figure 2-1), as follows:

- The exporting system will output the desired selection of data into an XML document. The resulting document must validate against the TransXChange schema version referenced in the document header.
- The document is transferred from the source to the target system by any appropriate transport method (e.g. email, ftp, and http).
- The importing system validates and imports the document, using the appropriate version of the TransXChange schema indicated by the document to interpret the document’s contents. It will reject the document if it is not well-formed (including the rules for internal integrity). It may decide its own actions to handling errors in the conforming to application level integrity constraints.

2.4.4 General Purpose Exchange of Data changes

A variant of the TransXChange general schema can be used to exchange ‘deltas’ just the changes since a previous exchange of data. A baseline version number and a “Changes since” date value can be used to indicate the relation to the previous release:
2.5 Differences between the Schemas

The TransXChange Registration and General schema are essentially the same, but differ in a few constraints as to cardinality and the required use of certain elements.

Table 2-1 summarises the differences between the two schema variations:

<table>
<thead>
<tr>
<th>TransXChange Registration Document</th>
<th>TransXChange General Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Must have a single Registration.</td>
<td>Can have zero or multiple Registration instances.</td>
</tr>
<tr>
<td>The Registration. Must reference a primary Service which describes the service being registered. Other connecting services to which the primary service connects can be included.</td>
<td>May have zero, one or many services.</td>
</tr>
<tr>
<td>The Service for the Registration must have a fully completed Registered Operator, i.e. of type LicensedOperator.</td>
<td>Registered operator details need not all be completed, i.e. can be of type Operator, rather than LicensedOperator.</td>
</tr>
<tr>
<td>Sufficient information about each stop must be provided to constitute a stand-alone definition for statutory purposes.</td>
<td>Simple Stop references may be used.</td>
</tr>
<tr>
<td>The Route information for the registered Service should include additional mapping point information where appropriate (using the RouteLink / Track / Mapping elements) to make the route unambiguous when the stop and mapping points are followed in sequence over a map containing a road network description.</td>
<td>Mapping information is optional.</td>
</tr>
<tr>
<td>Primary LocationSystem used in a Registration document must be Grid.</td>
<td>Either WGS84 or Grid can be used for LocationSystem. The same system should be used for all references in a given document.</td>
</tr>
</tbody>
</table>

Table 2-1 – Differences between Schemas

The schemas share a common set of element types (Figure 2-2). As a general principle, the Registration schema is strictly substitutable with the General schema, that is, a valid Registration document will always validate against both schemas and can be used wherever a General document is used.

Schema structure

Figure 2-2 – Common Set of Types in TransXChange Schemas
3 SHORT TOUR OF THE TRANSXCHANGE ESSENTIAL MODEL

In this chapter, we provide an overview of the physical model underlying the TransXChange schemas. Unified Modelling Language (UML) diagrams are used to show the relationships between the most significant elements.

3.1 Representing a Bus Service in TransXChange

The TransXChange model has seven basic concepts: Service, Registration, Operator, Route, StopPoint, JourneyPattern, and VehicleJourney.

- A Service brings together the information about a registered bus service, and may contain two types of component service: Standard or Flexible; a mix of both types is allowed within a single Service.
- A normal bus schedule is described by a StandardService and a Route. A Route describes the physical path taken by buses on the service as a set of route links.
- A FlexibleService describes a bus service that does not have a fixed route, but only a catchment area or a few variable stops with no prescribed pattern of use.
- A StandardService has one or more JourneyPattern elements to describe the common logical path of traversal of the stops of the Route as a sequence of timing links (see later), and one or more VehicleJourney elements, which describe individual scheduled journeys by buses over the Route and JourneyPattern at a specific time.
- Both types of service have a registered Operator, who runs the service. Other associated operator roles can also be specified.
- Route, JourneyPattern and VehicleJourney follow a sequence of NaPTAN StopPoints. A Route specifies in effect an ordered list of StopPoints. A JourneyPattern specifies an ordered list of links between these points, giving relative times between each stop; a VehicleJourney follows the same list of stops at specific absolute passing times. (The detailed timing Link and elements that connect VehicleJourneys, JourneyPatterns etc to StopPoints are not shown in Figure 3-1). StopPoints may be grouped within StopAreas.
- The StopPoints used in a JourneyPattern or Route are either declared locally or by referenced to an external definition using an AnnotatedStopRef
- A Registration specifies the registration details for a service. It is mandatory in the registration schema.

Figure 3-1 introduces, in UML class diagram notation, the core elements of the TransXChange schema. Reusable elements with a global scope are organized beneath the root TransXChange.
Figure 3.2 shows further elements of the TransXChange model.

- **A Calendar** may be specified that defines OperatingDays, DayTypes, and assignments between them.
- **A ServicedOrganisation** can be used to specify a school, works or other organisation served by a Service.
- **A StopPoint** may be part of a group of stops making up a StopArea, and may reside in a topographic region specified by an NptgLocality. Localities may be declared locally, or by reference to an external definition using an AnnotatedNptgLocalityRef.
- **A Route** may be made up of reusable RouteSections.
- **A JourneyPattern** may be made up of reusable JourneyPatternSections.
- **A Registration** may be accompanied by SupportingDocuments that pertain to it. Other SupportingDocuments may also be associated with the document as a whole.
- **DataRights** may be specified for the use of data elements.

**Figure 3-2 – UML Diagram of Elaboration of TransXChange model**

### 3.1.1 The NaPTAN Stop Model

TransXChange uses the NaPTAN stop model to define the stops and timing points of routes, and to associate stops with topographical locations in the National Public Transport Gazetteer (NPTG). For further details refer to the ‘NPTG and NaPTAN Schema Guide’.

Normally in TransXChange, stops comprise just a reference to an existing NaPTAN definition using a stop code; all such references are declared as AnnotatedStopPointRef instances. However, full StopPoint definitions for new bus stops may also be provided locally in a TransXChange document, using the NaPTAN StopPoint elements within the document. Each new locally defined stop definition must be allocated a NaPTAN identifier (that is an AtcoCode) that can be used to reconcile them with the NaPTAN database later.

#### 3.1.1.1 The NaPTAN Stop Model Introduction
Figure 3-3 summarises, in UML class diagram notation, the main stop elements of the TransXChange schema.

Stops are described using three main elements:

- **StopPoint**: Describes a stop, it contains a place, which is used to associate the stop with an NptgLocality: localities are defined in the NPTG database and are open to the Local Transport Authority to edit. Stops may be of a number of different types and subtypes, each with different properties.
  - **OnStreet / Bus**: MarkedStop, UnmarkedStop, HailAndRideSection, FlexibleZone.
  - **OffStreet / BusAndCoach**: Bay, VariableBay.
- **StopArea**: Used to group stops together.
- **NptgLocality**: Representing a topographical locality in the country, such as a city, town or village. Localities must exist in the NPTG database. Used to specify where a StopPoint or StopArea is relative to towns and cities.
- **AdministrativeArea**: All NaPTAN and NPTG elements are assigned to an administrative areas – this represents the organisation responsible for maintaining the stop data. See NaPTAN schema guide for further details.

*StopPoints* may be declared as either a *StopPoint*, or *AnnotatedStopPointRef*, indicating that further details may be found in the NaPTAN database. The latter is the normal mechanism.

![UML Diagram of Summary of Stop Model](image-url)
3.1.1.2 The NaPTAN Stop Model Details

Figure 3-4 shows further details of the NaPTAN stop elements. A StopPoint definition includes a Place & Descriptor groups. A StopAvailability may specify when a stop is available.

Figure 3-4 – UML Diagram of selected NaPTAN Stop elements
3.1.2 Resolving NaPTAN Stop References

When importing TransXChange schedules, an importing application will normally attempt to find the StopPoint details in the NaPTAN database using the NaPTAN identifier, i.e. the AtcoCode, and if found may - depending on the application's purpose - use the database's definition of the stop details in preference to any local definitions. Only if no existing StopPoint definition is found, will the locally declared definition be used. See Table 3-1.

<table>
<thead>
<tr>
<th>TransXChange Document use of stop</th>
<th>NaPTAN database</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaPTAN StopPointRef</td>
<td>Resolve to NaPTAN</td>
</tr>
<tr>
<td>Local NaPTAN declaration</td>
<td>Resolve to NaPTAN</td>
</tr>
<tr>
<td></td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>Use Local definition</td>
</tr>
</tbody>
</table>

Table 3-1 – Resolving Stop References

3.1.3 Variable Stop Allocations

For bus stations where the allocation of stops may vary over time, TransXChange supports variable stop allocation. In such cases the journey pattern should reference a NaPTAN stop of type BCQ, representing an unspecified stop or bay within the bus station, and then also specify a schedule of allocations to individual bays (i.e. NaPTAN stops of type BCT) for a given date, using the VariableStopAllocations element.
3.1.4 Stop Types

Every NaPTAN StopPoint has a stop type that indicates its mode and nature, for example, "on street, bus stop, marked ". Figure 3-5 shows, in UML class diagram notation, the stop classification elements of the NaPTAN schema. The main items of interest for TransXChange are:

- **OffStreet / BusAndCoach**, for stops in coach stations.
- **OnStreet / Bus** for stops on the street.

![Figure 3-5 – UML Diagram of Stop Classification Model](image-url)
3.1.5 NaPTAN Bus Stop Types

Figure 3-6 shows, in UML class diagram notation, the *BusStopType* elements of the NaPTAN schema.

![UML Diagram of On-street bus Stop Classification Model](image-url)
3.1.6 NPTG Administrative Support Types

*Figure 3-7* shows, in UML class diagram notation, the base types used by administrative elements of the *NPTG* schema.

![Diagram of NPTG Administrative Support types](image)

**Figure 3-7 – UML Diagram of NPTG Administrative Support types**

3.1.7 NPTG Locality Support Types

*Figure 3-8* shows, in UML class diagram notation, the Locality types used by elements of the *NPTG* schema.
3.1.8 NaPTAN Support Types

Figure 3-8 shows, in UML class diagram notation, the base types used by elements of the NaPTAN schema.

![Figure 3-8 – UML Diagram of NPTG Locality Support types](image-url)
Figure 3-9 – UML Diagram of NaPTAN Support types
3.1.9 NaPTAN Location Support Types

Figure 3-10 shows, in UML class diagram notation, the location data types used by the NPTG, NaPTAN & TXC schema.

![UML Diagram of NaPTAN Location Support Types](image-url)

Figure 3-10 – UML Diagram of NaPTAN Location Support Types
3.2 The Route and Service Supply Model

TransXChange describes a bus service using a model made up of three distinct layers or ‘levels of discourse’ (see Figure 3-11 for an UML diagram):

1. A Route; described as a sequence of route links connecting individual stops. For TransXChange, all stops are defined as being NaPTAN points, so a route describes a path in ‘NaPTAN space’; a distinct frame of reference made up of Public Transport Access Nodes (PTANs), which is semantically distinct from any given coordinate system, but which can be projected onto geospatial coordinate systems and mapping layers using Track elements.
   - The RouteLink instances are grouped using a RouteSection, allowing the reuse of whole sequences of links in different routes.
   - Track elements record both the plot of the route at non-NaPTAN points, and associations with mapping layer identifiers, such as OS TOIDS.

2. A JourneyPattern: a path over the route made up of a number of journey pattern timing links, each with timing information (and other optional operational data) ascribed to them. All timing information is relative (for example, ‘+5 minutes’).
   - Each end of a JourneyPatternTimingLink can have stop usage information associated with it on a JourneyPatternStopUsage element, specifying the activity at stop, and other service information.
   - The timing links are grouped using a JourneyPatternSection, allowing the reuse of whole sequences of links in different patterns.
   - The links of a JourneyPattern must traverse the same stops in the same sequence as the links of any Route associated with the JourneyPattern. However a JourneyPattern need not cover the whole Route; it may project onto just a contiguous subset of the links of the route, omitting route links at either or both ends.

   - Each vehicle journey has an absolute start time (e.g. ‘13:02’) specified: this can be combined with the timing information from each timing link to derive the actual passing times of departure and arrival at each timing point.
   - The public identifier of a VehicleJourney is given by a Line. One or more Line instances may be associated with a service, and a VehicleJourney must reference one of its service’s lines.
   - The link sequence of a VehicleJourney must exactly correspond to the link sequence of the underlying JourneyPattern; that is, each VehicleJourneyTimingLink must project onto a corresponding JourneyPatternTimingLink.

The Transmodel principles underlying the TransXChange Route and Service Supply model are summarised in Section 13.1, and divergences from Transmodel usage are listed.
Figure 3-11 – UML Diagram of Route, JourneyPattern and VehicleJourney Models
3.2.1 Model Layer Concerns

*Figure 3-12* illustrates how the each layer has a separate concern of the model:

1. The **Route** describes the stops, stop sequence, and the physical track between them.
2. The **JourneyPattern** adds in timing information; how long each link takes to run, how long to wait at each stop, and the allowed activities at each stop.
3. A **VehicleJourney** specifies a start time: this is used to compute actual passing times for each stop in the journey pattern, taking into account the run and waiting times. The vehicle journey can override the run time, wait time and activity from the journey pattern values for its own journey, but not change the stop sequence.

---

![Service Model Layers Diagram](image)
3.2.2 Summary of Route & Supply Model Elements

Each of the three layers is made up of three sets of broadly equivalent elements:

(i) Ordered collections, i.e. sequences, of links (Patterns and Sections).
(ii) Links (Route Links and Timing Links).
(iii) Link ends (Stop Usages).

Table 3-2 summarises the route and supply model elements, showing the simple one-to-one correspondences between equivalent elements in the different layers. The simple correspondence makes it straightforward to project between the route, journey pattern and vehicle journey layers. There are explicit references between elements in the pattern and link columns, which can be used to derive an implicit projection of the section and stop usage.

<table>
<thead>
<tr>
<th>Ordered Link Sequence</th>
<th>Link</th>
<th>Link end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route</td>
<td>RouteSection</td>
<td>RouteLink</td>
</tr>
<tr>
<td>(AbstractJourneyPattern) -</td>
<td>AbstractTimingLink</td>
<td>AbstractStopUsage</td>
</tr>
<tr>
<td>JourneyPattern</td>
<td>JourneyPatternSection</td>
<td>JourneyPatternTimingLink</td>
</tr>
<tr>
<td>VehicleJourney</td>
<td>VehicleJourneyTimingLink</td>
<td>VehicleJourneyStopUsage</td>
</tr>
</tbody>
</table>

Table 3-2 – Correspondence between Links and Nodes

3.2.3 Projection between Levels of Discourse

Figure 3-13 shows a schematic example of links at different levels of discourse and the correspondences between them.

Figure 3-13 – Correspondence between Links at Different Levels
3.3 Route Model

3.3.1 Introduction to the Route Model

Figure 3-14 introduces the elements used to represent a Route. These comprise Routes, RouteSections (Reusable sequences of RouteLinks), and RouteLinks. The projection of the route is described by a Track made up of Instruction and Feature elements.
3.3.2 Route Model Details

Figure 3-15 elaborates Figure 3-14 to show further attributes.

Figure 3-15 – UML Diagram of Route Model

3.3.3 Tracks

The TransXChange Track model describes details about the physical course of a RouteLink, in particular the collection of spatial points needed to plot the route unambiguously in sequence on a map of the road network, for example using a ‘snap to track’ algorithm. As well as such a Mapping, a Track can also be associated with a reference to an external mapping system using a MapSystemReference element, allowing the projection of links onto geospatial map layers. Track features can also be used to describe any manoeuvre involved in navigating a route link, such as a U-turn.

The Track model allows a rich description of a route to be provided; it is intended for general purpose data exchange. For a Registration a level of Track detail should be given sufficient to unambiguously plot the route on a map using OSGR data – using both points and/or TOIDS.
It is a requirement of registration that adequate spatial data is provided as to plot routes on an OS map in a useful way: there should be intermediate coordinates for a reasonably high level of resolution.

*Figure 3-14* shows a UML structure diagram of the elements used to describe tracks. Tracks can contain two different types of description:

- A **Mapping** describes the geospatial plot of the route link as two or more **Location** elements that provide point coordinates for the track between NaPTAN stop points.
- An **Instructions** instance provides an optional additional structured description of the steps involved in traversing the track as a sequence of **Feature** instances. For example ‘Turn left at roundabout into Mary Street’.

### 3.3.4 Track Example

As a simple example, consider a **RouteLink** that runs along the B205 and B257, represented by a two **Track** instances.

- Each **Track** instance has a **Mapping** instance that describes the course of the track. tr1 has two points \((g_1, g_2)\) and tr2 has seven points \((g_3 \text{ to } g_7)\) respectively; each point is a **Location** instance that describes a point of the track.

- Each **Track** has an **Instructions** instance containing an ordered collection of **Feature** instances.

- Each **Feature** instance describes a step needed to traverse the track, and references a **Location** instance from the Table 3-3 shows a sample of the **Feature** instances.

<table>
<thead>
<tr>
<th>Track</th>
<th>Location Ref</th>
<th>Feature Type</th>
<th>Relative Bearing</th>
<th>Absolute Bearing</th>
<th>Onward Name</th>
<th>Road Number</th>
<th>Distance</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tr1</td>
<td>G_1</td>
<td>legOrigin</td>
<td>straightAhead</td>
<td>N</td>
<td>Victoria Road</td>
<td>B205</td>
<td>300m</td>
<td>Proceed 300m North down Victoria road (B205.)</td>
</tr>
<tr>
<td></td>
<td>G_2</td>
<td>junction</td>
<td>left</td>
<td>W</td>
<td>Albert Road</td>
<td>B205</td>
<td>500m</td>
<td>Turn left into Albert road (B257) and head west 500m.</td>
</tr>
<tr>
<td>Tr2</td>
<td>G_3</td>
<td>landmark</td>
<td>straightAhead</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>Hospital on left</td>
</tr>
<tr>
<td></td>
<td>G_4</td>
<td>bend</td>
<td>right</td>
<td>NW</td>
<td>Albert Road</td>
<td>B257</td>
<td>--</td>
<td>Follow bend to right in Albert Road</td>
</tr>
<tr>
<td></td>
<td>G_5</td>
<td>roadChange</td>
<td>straightAhead</td>
<td>NW</td>
<td>George Road</td>
<td>B257</td>
<td>400m</td>
<td>Continue 400m down George Road (B257)</td>
</tr>
<tr>
<td></td>
<td>G_6</td>
<td>roundabout</td>
<td>left</td>
<td>SW</td>
<td>Mary Street</td>
<td>B257</td>
<td>--</td>
<td>Turn left at roundabout into Mary Street</td>
</tr>
<tr>
<td></td>
<td>G_7</td>
<td>crossing</td>
<td>straightAhead</td>
<td>--</td>
<td>Bill Alley</td>
<td>B257</td>
<td>--</td>
<td>Cross over Bill Alley</td>
</tr>
<tr>
<td></td>
<td>G_8</td>
<td>bridge</td>
<td>straightAhead</td>
<td>--</td>
<td>Mary Street</td>
<td>B257</td>
<td>--</td>
<td>Pass under bridge</td>
</tr>
</tbody>
</table>
### 3.3.5 Route Model Hierarchy

Figure 3-16 shows the inheritance hierarchy for the Route model elements.

**Table 3-3 – Example Track Instructions**

<table>
<thead>
<tr>
<th>legDestination</th>
<th>straightAhead</th>
<th>S</th>
<th>Mary Street</th>
<th>B257</th>
<th>600m Continue straight ahead 600m South down Mary Street</th>
</tr>
</thead>
</table>

**Figure 3-16 – UML Diagram of Route Model Element Hierarchy**
3.4 Journey Pattern Model

3.4.1 Introduction to the Journey Pattern Model

Figure 3-24 introduces the elements used to represent a JourneyPattern. These comprise Journey-Patterns, JourneyPatternSections (Reusable ordered sequences of JourneyPatternTiming-Links), and JourneyPatternTimingLinks. Each JourneyPatternTimingLink connects two stops: attributes for each link end may be specified by a JourneyPatternStopUsage.
### 3.4.2 Journey Pattern Model Details

*Figure 3-18* elaborates *Figure 3-24* to show the detailed attributes of *JourneyPattern* elements.
3.4.3 Journey Pattern Model Hierarchy

Figure 3-19 shows the inheritance hierarchy for the JourneyPattern model elements.

Figure 3-19 – UML Diagram of Journey Pattern Model Element Hierarchy
3.5 Vehicle Journey Model

3.5.1 Introduction to the Vehicle Journey Model

Figure 3-24 introduces the elements used to represent a VehicleJourney. These comprise AbstractVehicleJourney, VehicleJourney, ConnectingVehicleJourneys (Lightweight VehicleJourneys for describing external interchanges) and VehicleJourneyTimingLinks.

Each VehicleJourneyTimingLink connects two stops: attributes for each link end may be specified by a VehicleJourneyStopUsage. A Frequency element further describes the intervals of Frequency Based VehicleJourneys. A DeadRun (see separate section later) describes a vehicle positioning run that does not appear in the public timetable.
Figure 3-20 – UML Diagram of Journey Pattern Model: Introduction
3.5.2 Vehicle Journey Model Details

Figure 3-15 elaborates Figure 3-14 to show the attributes of VehicleJourney elements.

Figure 3-21 – UML Diagram of Vehicle Journey Model: Details
3.5.3 Connecting Vehicle Journey Model

Figure 3-16 shows the model for a ConnectingVehicleJourney. A connecting vehicle allows details of a connecting journey to be referenced.

Figure 3-22 – UML Diagram of Connecting Vehicle Journey Model
3.5.4 Vehicle Journey Model Hierarchy

Figure 3-16 shows the inheritance hierarchy for the **VehicleJourney** model elements.

Figure 3-23 – UML Diagram of Vehicle Journey Model Element Hierarchy
3.6 The Use of Links in TransXChange

In Transmodel, a journey can be regarded either as an ordered list of stops, or as an ordered list of links between the stops: both views can be derived from the underlying TransXChange representation of a journey pattern and vehicle journey as a list of timed links. In TransXChange, a ‘timing link in link sequence’ representation is used (see discussion of Transmodel terminology and concepts in section 13.2), as this holds more information than a simple stop list, and can be projected exactly onto a spatial route; it can readily be transformed by applications into a list of stops and passing times if needed.

The following Transmodel principles apply to the use of journey patterns in TransXChange:

1. There should be a separate journey pattern for each physical route followed, i.e. a sequence of timing links between stops defining a unique sequence of stops.

2. A vehicle journey must always follow a journey pattern.

3. A vehicle journey must visit all the stops of a journey pattern, with two qualifications (which are not strictly Transmodel - see 13.2):
   a. Short working of the underlying journey pattern is allowed, i.e. truncation of one or more stops at either or both ends.
   b. Express journeys over a service pattern are allowed – i.e. provided a journey traverses a link and goes past a stop, it may specify an activity of ‘pass’ to omit a particular stop.

The following further principles apply to the use of links to represent journey patterns in TransXChange:

4. A vehicle journey need specify explicitly only those timing links that are different from the underlying journey pattern. Other vehicle journey links may be implicit, that is derived automatically from the underlying journey pattern. In many cases, no explicit concrete links need be specified in a vehicle journey.

5. A vehicle journey may reference all the links of another vehicle journey. In this case all the link usage must be implicit, that is, all of the links of the referenced journey are used with the same values as in the referenced journey. If the vehicle journey needs to make modifications to links or link properties, it should be based directly on an underlying journey pattern, and not reference another vehicle journey for some links and make further changes.

6. Timing links may have a number of different ‘successive’ properties that change over successive steps of the journey pattern, for example, destination headings, duty crews, and fare stages. The properties may be set on individual links at both the journey pattern and vehicle journey level. Once a successive property (such as a dynamic destination heading) is set on a specific link (or individual link end), it is considered to be in effect on successor links in the journey until any different value is encountered on a subsequent link. Link values on successor vehicle journey links may either be set explicitly, or be inherited from a parent journey pattern link.
3.6.1 Structure Example of a Schedule with one Pattern and Two Journeys,

*Figure 3-24* shows a simple route, with five stops connected by four links.

*Figure 3-24 – Simple Route Map*

*Table 3-4* shows an example timetable of a service running over the route, with two vehicle journeys running between each of the five stops.

<table>
<thead>
<tr>
<th>Name/Line</th>
<th>A</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grub Street</td>
<td>d</td>
<td>8:02</td>
</tr>
<tr>
<td>Tin Pan Alley</td>
<td>d</td>
<td>8:12</td>
</tr>
<tr>
<td>Sinister Street</td>
<td>d</td>
<td>8:37</td>
</tr>
<tr>
<td>Swans Way</td>
<td>d</td>
<td>8:45</td>
</tr>
<tr>
<td>Howard’s End</td>
<td>a</td>
<td>8:55</td>
</tr>
</tbody>
</table>

*Table 3-4 – Structure Example of a Schedule*

*Table 3-5* shows this same timetable annotated with the XML element instances needed to represent it in a *TransXChange* XML document.

- The service has a single Line *Ln_1* with a Line Name of ‘A1’.
- The service is presented in a matrix of five rows of stops (*S_1 – S_5*), and two columns of journeys (#1 – #2), each column showing a vehicle journey stopping at each row.
- There is one route (*R_1*), with a single route section (*RS_1*) of four route links (*RL_1, RL_2, RL_3, and RL_4*). Each route link has two stop references (*RL_1a, RL1b*, etc).
- The service is made up of a single journey pattern (*JP_1*). The journey pattern, section and timing links correspond to those of the route; there is a single journey pattern section (*JS_1*), and four timing links (*JL_1, JL_2, JL_3, JL_4*), with individual run times of 10, 20, 8, and 10 minutes respectively. (There is also a 5 minute wait at sinister street.
  - Each journey pattern timing link has two stop usages (*JL_1a, JL_1b*, etc) for each end of the link, i.e. on for departure, one for arrival. These can hold information about the use of the stop.
- There are two vehicle journeys (*VJ_1, VJ_2*), that both use the same pattern, and that are for the same line, ‘A1’ (*Ln_1*).
  - For *VJ_1*, each of the four vehicle journey timing links (*VL_1, VL_2, VL_3, VL_4*) corresponds to a link of the journey pattern, and has its own pair of stop usages (*VL_1a, VL_1b*, etc).
  - Times at each stop are computed from the vehicle journey start time (e.g. ‘8.02’) and the individual link run times (e.g. +10mn), plus any wait time on the stop usage. (For *S_1 – S_4*, only departure times are actually shown in Table 3-5; for *S_5* it is the arrival time).
  - The second vehicle journey *VJ_2* reuses the links of the first journey pattern *VJ_1*, with a different start time (‘10:02’).
3.6.2 Structure Example of a Schedule with an Express Journey

As a slight variation on the structure example given above, we consider a second example (Table 3-6), in which the second vehicle journey (VJ_3) omits a particular stop (S_2) in the same journey pattern (JP_1).

- The second journey declares its own distinct set of vehicle journey timing links (VL_3_1, VL_3_2, VL_3_3, and VL_3_4) for the journey, so that it can modify the activity. These are based on the same journey pattern.
- For the stop that is omitted (S_2), an override value of ‘pass’ is specified for the activity on the vehicle journey stop usage of the link ends which connect to the stop (VL_3_1b, VL_3_2a).

<table>
<thead>
<tr>
<th>Route Section</th>
<th>Section</th>
<th>VJ_1</th>
<th>VJ_2</th>
<th>Name/Line</th>
<th>A1</th>
<th>A1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Ref</td>
<td>Link</td>
<td>Usage</td>
<td>Link</td>
<td>Usage</td>
<td></td>
</tr>
<tr>
<td>S_1</td>
<td>RL_1</td>
<td>RL_1a</td>
<td>JL_1</td>
<td>JL_1a</td>
<td>VL_1a</td>
<td>8:02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Grub Street</td>
<td>8:02</td>
</tr>
<tr>
<td>S_2</td>
<td>RL_2</td>
<td>RL_2a</td>
<td>JL_2</td>
<td>JL_2a</td>
<td>VL_2a</td>
<td>8:12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tin Pan Alley</td>
<td>8:12</td>
</tr>
<tr>
<td>S_3</td>
<td>RL_3</td>
<td>RL_3a</td>
<td>JL_3</td>
<td>JL_3a</td>
<td>VL_3a</td>
<td>8:37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sinister Street</td>
<td>8:37</td>
</tr>
<tr>
<td>S_4</td>
<td>RL_4</td>
<td>RL_4a</td>
<td>JL_4</td>
<td>JL_4a</td>
<td>VL_4a</td>
<td>8:45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Swans Way</td>
<td>8:45</td>
</tr>
<tr>
<td>S_5</td>
<td>RL_5</td>
<td>RL_5a</td>
<td>JL_5</td>
<td>JL_5a</td>
<td>VL_5a</td>
<td>8:55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Howard’s End</td>
<td>8:55</td>
</tr>
</tbody>
</table>

Table 3-6 – Structure Example of Schedule: Express VehicleJourney

3.6.3 Plotting a route on a Map

If Track data is present it can be used to plot an exact route track on a map. In this case the Mapping data should be regarded as independent of the stop locations. That is to plot a route the last point of each mapping is connected to the first point of the succeeding Mapping. Thus the track data may follow the centreline of the road.
### 3.7 Inheriting Timing Link Values

*Table 3-7 shows the various values that may be specified for the `VehicleJourney` and `VehicleJourneyTimingLink` elements, and whether they are:

(i) **Required** ['R'].
(ii) **Optional but otherwise inherited** from the previous level of discourse ['O'].
(iii) **Always inherited** ['I'].
(iv) The most significant properties are the actual run and wait times of each timing link, but several other operational values, such as fare stages, may also be specified.
For elements that are optional at all levels, a default value is identified to use if no explicit value is provided. For some ‘successive’ properties, such as fare stage number, the value in effect from any previous link is assumed unless specified otherwise. This is indicated by a ['S'].*

<table>
<thead>
<tr>
<th>Level</th>
<th>Property</th>
<th>Service</th>
<th>Route</th>
<th>Journey Pattern</th>
<th>VehicleJourney</th>
<th>Default Value</th>
<th>TXCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern</td>
<td>ServiceRef</td>
<td>--</td>
<td>(R)</td>
<td>I</td>
<td></td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Direction</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>Outbound</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>OperatorRef</td>
<td>R</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>Service / RegisteredOperator</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>DestinationDisplay</td>
<td>(R)</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>Service /Destination</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>TicketMachineServiceCode</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>TicketMachine / JourneyCode</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TicketMachine / Direction</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>Direction</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Block / Board</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Block / BoardNumber</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Block / Note</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GarageRef</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VehicleType</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>AssistanceService</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>ServiceFacilitySet</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>LayoverPoint</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TimeDemand</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CommercialBasis</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>unknown</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Frequency</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>false</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OperatingProfile</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>Monday to Friday, Every Day of Year</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>LineRef</td>
<td>--</td>
<td>--</td>
<td>R</td>
<td>--</td>
<td></td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>DepartureTime</td>
<td>--</td>
<td>--</td>
<td>R</td>
<td>--</td>
<td></td>
<td>2.1</td>
</tr>
<tr>
<td>Section</td>
<td>order</td>
<td>--</td>
<td>O</td>
<td>I</td>
<td>None</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>TimingLink</td>
<td>LinkRef</td>
<td>--</td>
<td>O</td>
<td>R</td>
<td>--</td>
<td></td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Direction</td>
<td>O</td>
<td>R</td>
<td>O</td>
<td>I</td>
<td>JourneyPattern / Direction</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>RunTime</td>
<td>--</td>
<td>R</td>
<td>O</td>
<td>--</td>
<td></td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Distance</td>
<td>O</td>
<td>O</td>
<td>I</td>
<td>zero</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DestinationDisplay</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>(same as Pattern / DestinationDisplay)</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Vias /ViaName</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HailAndRide</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>false</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DutyCrewCode</td>
<td>--</td>
<td>O</td>
<td>S</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>CommercialBasis</td>
<td>O</td>
<td>--</td>
<td>O</td>
<td>none</td>
<td>Same as pattern</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>StoppingArrangements</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>TimingLink</td>
<td>StopPointRef</td>
<td>(R)</td>
<td>R</td>
<td>I</td>
<td>--</td>
<td></td>
<td>2.1</td>
</tr>
<tr>
<td>StopUsage</td>
<td>TimingStatus</td>
<td>--</td>
<td>O</td>
<td>I</td>
<td>TIP</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td>From &amp; To</td>
<td>Activity</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>PickUpAndSetDown</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WaitTime</td>
<td>(++)</td>
<td>O</td>
<td>O</td>
<td>zero</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VariableStopAllocation</td>
<td>--</td>
<td>O</td>
<td>O</td>
<td>none</td>
<td>2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FareStageNumber</td>
<td>--</td>
<td>O</td>
<td>S</td>
<td>I</td>
<td>none</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>FareStage</td>
<td>--</td>
<td>O</td>
<td>I</td>
<td>false</td>
<td>2.1</td>
<td></td>
</tr>
</tbody>
</table>
++ A default wait time may be specified on stops. This merely sets a default that may be used to set the initial value used by services. Each journey pattern sets the wait value on each timing link.

Figure 3-25 shows how inheritance relationships are used in the TransXChange supply model so as to express the shared attributes and common data structure of equivalent elements, that is, the elements and subelements of JourneyPattern and VehicleJourney. For each element type, an abstract class is used to represent the common properties, and distinct subtypes describe any specific differences. For example, AbstractTimingLink has subtypes JourneyPatternTimingLink and VehicleJourneyTimingLink.

See Table 3-7 for attributes which may be inherited.

- A VehicleJourney may override any common property it shares with a JourneyPattern.
- A VehicleJourneyTimingLink may override any common property it shares with a JourneyPatternTimingLink.
- A VehicleJourneyStopUsage may override any common property it shares with a JourneyPatternStopUsage, including the StopAccessibility.

![Figure 3-25 – UML Diagram of Service Pattern elements](c) 2001-2010 Crown Copyright)
3.7.1 Schedule and Journey Terms and Definitions

The TransXChange uses the following definitions of common scheduling terms. See also the definitions of individual schema elements. Some of the terms are used in actually element names; others merely define concepts.

3.7.1.1 Time Related Terms

- **Relative time**: Time as a duration, usually in minutes, for example, ‘5 minutes’.
- **Absolute time**: Time as a specific clock hour, for example, ‘10:00’, ‘18:30’.
- **Overall Wait time**: Relative time to wait at a specific stop, assuming bus arrives on time. Used to compute passing times. In real-time operations, if bus is late at a stop, wait time may be reduced to the minimum time need to disembark and board passengers, i.e. wait is a buffer time used to adhere to schedule. The actual time waited is the *Dwell time* - which is an operational time and not relevant to TransXChange. Note that wait time is a property of a journey pattern or vehicle journey, not of the stop itself, since it may be different on different journeys using the same stop. In TransXChange, the overall wait time is computed from two separate component timing link wait times that can be stated on each end of the incoming and outgoing *JourneyPatternTimingLink* or *VehicleJourneyTimingLink* instances:
  - See *JourneyPatternStopUsage / WaitTime*.
  - See *VehicleJourneyStopUsage / WaitTime*.
- **Run time**: Relative time taken to traverse a timing link.
  - See *JourneyPatternTimingLink / RunTime*.
  - See *VehicleJourneyTimingLink / RunTime*.
- **Departure Time**: The absolute time at which a vehicle journey leaves from its first stop.
  - See *VehicleJourney / DepartureTime*.
- **Passing time**: Absolute time that a bus reaches a stop. Comprises the departure time from the previous stop, plus the run time for the timing link connecting the previous stop and the next stop. Derived.
- **Frequency Based Service**: A service that runs to a regular frequency, for example ‘every 5 minutes’, rather than to a specific timetable. May or may not be a strict Frequent Service.
  - See *VehicleJourney / Frequency*.
- **Frequent Service**: A service that runs to a frequency of every 10 minutes or less in accordance with the Statutory Requirement, and that has been formally registered as constituting a Frequent Service. Normally, but not necessarily, a Frequency Based Service.
  - See *VehicleJourney / Frequency / FrequentService*.
- **Day Type**: A type of day or day such as Monday, Weekday, or Weekend as opposed to a calendar date.

3.7.1.2 Routing Related Terms

- **Block**: A description of a group of journeys to be operated by a particular vehicle, in a specific working period, normally covering a full working day. May be identified by a *block number*.
  - See *JourneyPattern / Block / Description*.
  - See *JourneyPattern / Block / BlockNumber*.
- **Origin**: The place from which the service starts. Does not vary; note however that some journeys of the service may have a ‘short working’ so start from a different actual origin when executed.
  - See *Service / Origin*.
- **Destination**: The place to which the service goes. Does not vary; Note however that some journeys of the service may have a ‘short working’ and so go to a different actual destination when executed.
  - See *Service / Destination*.
- **Destination Display**: Name of a destination to which the bus ultimately goes. Fixed for whole journey.
  - See *JourneyPattern / DestinationDisplay*.
  - See *VehicleJourney / DestinationDisplay*.
- **Dynamic Destination Display**: Name of a destination where the bus is currently considered heading, shown on the front of the bus. Also known as the *Heading*. On a circular or other route with a complex topology, the destination display may change from stop to stop. On a
linear route, normally the same as the destination display, but on a short working may be an earlier point in the pattern.
  o See JourneyPatternTimingLink / DestinationDisplay.
• Stop List. The actual list of stops at which the bus will stop, in order of visiting. Sometimes also termed the ‘calling pattern’.
  o See JourneyPattern / Direction, JourneyPatternTimingLink / Direction.
• Via List: The list of place names that is published for the service. This may be a subset of the stop list and may include names that are not stops
  o See JourneyPatternTimingLink / DestinationDisplay.
• Direction: relative course of a bus following a vehicle journey – may be outbound, inbound, clockwise or anti-clockwise.
  o JourneyPattern / Direction, JourneyPatternTimingLink / Direction.
• Bearing, Absolute, i.e. compass direction of a bus along a street, e.g. ‘North’.
  o See StopPoint / Bearing.
• Layover Point: Point at which a bus may stop and wait until it is time to start the next service stage.
  o See JourneyPattern / LayoverPoint.
• Short Working: A vehicle journey that follows a journey pattern but omits one or more stops at one or other end of the journey.
  o See VehicleJourney / DeadRun / EndStopUsage.
• Express Journey: A vehicle journey that follows a journey pattern but passes certain stops without stopping (also referred to as a Limited Stop Journey).
  o See JourneyPatternTimingLink / Activity.
• Stop Footprint: The geometry of the stop coverage. Most stops are points. Some stop types however have a footprint that covers more than a single point, for example hail and ride sections, or flexible zones.

3.7.2 Computation of Passing Times

The passing time at each stop (see Figure 3-26) is calculated from the cumulative sum of the individual timing link values for all preceding stops in the journey link sequence as follows:

[1] Arrival time at stopₙ = Departure time from previous stopₙ₋₁ + (Run time for inbound link from stopₙ₋₁)

[2] Departure time at stopₙ = Arrival time at stopₙ + Wait time for destination end of inbound link from stopₙ₋₁, + Wait time for origin of outbound link to stopₙ₊₁

Where:

1. Default vehicle journey wait times for each link are derived from the journey pattern timing link onto which the vehicle journey timing link projects (i.e. through the VehicleJourneyTimingLink / JourneyPatternTimingLinkRef), as follows:
   • If no value for wait time is specified on the departure end of the timing link, i.e. for the VehicleJourneyTimingLink / From / VehicleJourneyStopUsage, the default WaitTime from the corresponding JourneyPatternTimingLink / From / JourneyPatternStopUsage is used.
   • If no value for wait time is specified on the arrival end of the timing link, i.e. the VehicleJourneyTimingLink / To / VehicleJourneyStopUsage, the default WaitTime from the corresponding JourneyPatternTimingLink / To / JourneyPatternStopUsage is used.

2. If unspecified, journey pattern wait times are defaulted as follows:
   • If no value for wait time is specified on the departure end of the timing link, i.e. the JourneyPatternTimingLink / From / JourneyPatternStopUsage, a value of zero is used.
3. If no value for wait time is specified on the arrival end of the timing link, i.e. the JourneyPatternTimingLink / To / JourneyPatternStopUsage, a value of zero is assumed.

3. Default vehicle journey run times for each link are derived from the journey pattern timing link onto which the vehicle journey timing link projects. A run time is mandatory on each JourneyPatternTimingLink.

The structured example shown earlier gives a simple example of how passing times are derived from run times and wait times.

**Timings**

<table>
<thead>
<tr>
<th>Arrival</th>
<th>Depart</th>
<th>Arrival</th>
<th>Depart</th>
<th>Arrival</th>
<th>Depart</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00</td>
<td>10:00</td>
<td>10:05</td>
<td>11:05</td>
<td>11:05</td>
<td>11:16</td>
</tr>
</tbody>
</table>

**Figure 3-26 – Computation of Passing Times**

3.7.2.1 Example of Inheritance of Passing Times

*Table 3-8* shows a more complex example, where wait and run times are specified at different levels of discourse, that is, default values from the journey pattern are used except where overridden by the vehicle journey. For each step, the wait and run times are added to values from the previous step to arrive at an overall passing time. There are three stops S1, S2, S3 and two links (L1, L2) between them.

- An initial time of ‘10:00’ is specified.
- Run time $R_1$ (5 minutes) on the vehicle journey pattern timing link (L1) is defaulted from the journey pattern timing link.
- Run time $R_2$ (10 minutes) on the vehicle journey pattern timing link (L2) overrides the default (14 minutes) on the journey pattern.
- Departure wait time $W_{1b}$ at S1 (2 minutes) on the vehicle journey timing link end L1a overrides the default (0 minutes) on the journey pattern.
- Arrival Wait time $W_{2a}$ (5 minutes) at S2 on the vehicle journey timing link end L1b is defaulted from the journey pattern.
- Departure wait time $W_{2b}$ at S2 (7 minutes) on the vehicle journey timing link end L2a overrides the default (6 minutes) on the journey pattern.
- Arrival Wait time $W_{3a}$ (10 minutes) at S3 on the vehicle journey timing link end L2b overrides the default (5 minutes) on the journey pattern.
- Departure wait time $W_{3b}$ at S3 (5 minutes) - which would come from a successor link L3) can be used to compute the departure time from S3.
3.7.2.2 Rounding of Passing Times

Run and wait times are specified as values of XML type Duration, which may include seconds, for example PT10M55S. The TransXChange publisher computes departure times using the full value including seconds, but in the matrix timetable rounds down the total cumulative time to the nearest whole minute, i.e. the rounded value is not used to reset the cumulative time. Table 3-9 gives an example.

### Table 3-8 – Example of Computation of Inherited Passing Times

<table>
<thead>
<tr>
<th>Stop</th>
<th>Wait Time</th>
<th>Link</th>
<th>Run Time</th>
<th>Computation</th>
<th>Passing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>JP</td>
<td>VJ</td>
<td>Actual id</td>
<td>JP mns</td>
<td>VJ mns</td>
</tr>
<tr>
<td>s1</td>
<td>-</td>
<td>-</td>
<td>--</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>w1b</td>
<td>From (+0)</td>
<td>0</td>
<td>+2</td>
<td>1</td>
<td>L1a</td>
</tr>
<tr>
<td>r1</td>
<td></td>
<td>L1</td>
<td>+5</td>
<td>--</td>
<td>+5</td>
</tr>
<tr>
<td>s2</td>
<td>w2a</td>
<td>To</td>
<td>+5</td>
<td>+5</td>
<td>L1b</td>
</tr>
<tr>
<td>w2b</td>
<td>From (+6)</td>
<td>+7</td>
<td>+7</td>
<td>2</td>
<td>L2a</td>
</tr>
<tr>
<td>r2</td>
<td></td>
<td>L2</td>
<td>(+14)</td>
<td>+10</td>
<td>+10</td>
</tr>
<tr>
<td>s3</td>
<td>w3a</td>
<td>To</td>
<td>+5</td>
<td>+10</td>
<td>L2b</td>
</tr>
<tr>
<td>w3b</td>
<td>From --</td>
<td>+5</td>
<td>+5</td>
<td>3</td>
<td>L3a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L3</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
</tbody>
</table>

### Table 3-9 – Example of Rounding of Passing Times

<table>
<thead>
<tr>
<th>Stop</th>
<th>Run Time</th>
<th>Cumulative Time</th>
<th>Show As</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>PT20M00S</td>
<td>7:00:00</td>
<td>7:00</td>
</tr>
<tr>
<td>B</td>
<td>PT20M50S</td>
<td>7:20:50</td>
<td>7:20</td>
</tr>
<tr>
<td>C</td>
<td>PT20M50S</td>
<td>7:41:40</td>
<td>7:41</td>
</tr>
<tr>
<td>D</td>
<td>PT10M55S</td>
<td>7:52:35</td>
<td>7:52</td>
</tr>
</tbody>
</table>
3.8 Standard Services

3.8.1 Introduction to the Service Model

Figure 3-27 introduces the TransXChange Service model which groups the journeys of a TransXChange document and associates them with common properties. Each Service can be made up of StandardService and FlexibleService components.

Figure 3-27 – UML Diagram of Service model: Introduction

3.8.2 Introduction to the Standard Service Model

Figure 3-28 gives a slightly more detailed view of the Service TransXChange model introduced above, summarising the overall structure of a StandardService, and showing again that JourneyPattern, and VehicleJourney are made up of collections of timing links (JourneyPatternTimingLink, and VehicleJourneyTimingLink respectively), which hold the details about each individual step between stops of the journey.

- Each timing link has information about the arrival and departure of the vehicle at a stop, specified with a stop usage element (JourneyPatternStopUsage, and VehicleJourneyStopUsage respectively).
- For Bus Stations, stop i.e. bay allocation may be variable, specified by a VariableStopAllocation.

A StandardService describes the fixed route component of a Service.

- Each Service can have one or more Line instances associated with it; this specifies a label to be associated with journeys, for example, “N93”.
- Each StandardService must have one or more JourneyPattern instances.
  - A JourneyPattern instance may reference a Route and a Track.
The **StandardService** must have one or more **VehicleJourney** instances. Each **VehicleJourney** instances must reference a **JourneyPattern** of the same **StandardService**, and a **Line** instance of the **Service** to which it belongs.

Each **VehicleJourney** must specify a **DepartureTime**: Frequency based services may also describe a **Frequency**. See 0 below.

Connections with other services are described by *interchanges*. These are described in Section 3.10.

---

**Figure 3-28 – UML Diagram of Standard Service: Overview**
3.8.3 Standard Service Properties

Figure 3-29 shows further details of a Service including a ServiceClassification.
3.8.4 Standard Service Components

Figure 3-30 shows further details of the **StandardService** and **FlexibleService** components of a Service.

![Figure 3-30 – UML Diagram of Standard Service Parts](image-url)
3.8.5 Service Model Hierarchy

Figure 3-31 shows the inheritance hierarchy for the Service model elements.

Figure 3-31 – UML Diagram of Service Model Element Hierarchy
3.8.6 Service Model Types

Figure 3-32 shows the additional data types used the Service model elements.

![Diagram of Service Support types]

**Figure 3-32 – UML Diagram of Service Support types**
3.9    Flexibly Routed Services 

The TransXChange model can also support flexibly routed services (Figure 3-33). A flexible service operates between catchment areas that can be made up of both spatial zones, and lists of fixed stops, allowing combinations of (i) area-to-fixed stop, (ii) area-to-area, (iii) fixed stop-to-fixed stop. Within a zone there is no fixed or marked stop, but the service will call on demand.

Figure 3-33 – Flexible Network

3.9.1 Introduction to the Flexible Service Model

Figure 3-34 introduces the elements used to represent a FlexibleService.
A **FlexibleService** has a **FlexibleJourneyPattern**, which must include some NaPTAN stops of type **FlexibleZone** (FLX) to define areas within which passengers may be picked up or set down.

- **FlexibleZone** must be a contiguous area. Like other NaPTAN stop types, a **FlexibleZone** stop can be associated with one or more NPTG Localities: the locality with the greatest correspondence to the area of the zone should be used as the primary NPTG Locality; other localities that the zone falls within should be specified as alternative NPTG localities on the NaPTAN stop definition. Where a flexible zone substantially covers two or more NPTG Localities, it is preferable to define two separate zones, one for each locality.

- **FlexibleJourneyPattern** may also have one or more **FixedStopPoint** instances that can be visited in any order by the flexible service. Fixed stops should be NaPTAN stops of a type other than **FlexibleZone** (FLX).

- The allowed activity (pick up, set down etc) and other behaviour of the service at each stop, fixed or flexible, is defined by a stop usage instance for each stop used.

- A **FlexibleVehicleJourney** describes the actual operation of the flexible service, using a **FlexibleServiceTimes** element to specify the time bands during which the service operates.

- A **Service** may contain both **FlexibleService** and **StandardService** components. Interchange elements can be used to define the transition between flexible and fixed stages.

- Other properties of the service, such as **Registration**, **Operator**, **Line** and **OperatingProfile**, are specified with the same elements as for a **StandardService**.

---

![Figure 3-34 – UML Diagram for Flexibly Routed Service: Introduction](image-url)
3.9.2 Flexible Service Model Details

*Figure 3-35 elaborates Figure 3-34 to show the attributes of Flexible Service elements.*

---

**Figure 3-35 – UML Diagram for Flexibly Routed Service: Details**

3.9.3 Flexibly Routed Service Hierarchy

*Figure 3-36 shows the inheritance hierarchy for the Flexibly Routed Service model elements.*
Figure 3-36 – UML Diagram of Flexibly Routed Service Element Hierarchy
3.10 Line Model

3.10.1 Introduction to the Line Model

In TransXChange a Line is an arbitrary grouping of journeys under a common public identifier. Each service may have multiple lines. Figure 3-37 introduces the elements used to represent a Line. These comprise Line and up to two LineDescription instances. Each VehicleJourney may reference a Line.

Figure 3-37 – UML Diagram of Line Model: Introduction
3.10.2 Line Model Details

Figure 3-38 elaborates Figure 3-37 to show further attributes.

![UML Diagram of Line Model](figure3-38.png)

3.10.3 Line Hierarchy

Figure 3-39 shows the inheritance hierarchy for the Line model elements.

![UML Diagram of Line Model Element Hierarchy](figure3-39.png)
3.11 Interchanges

To specify the connection between vehicle journeys, an Interchange model is used, as shown in the UML structure diagram in Figure 3-40. The Interchange model operates on two levels of discourse:

- A *JourneyPatternInterchange* specifies a possible connection between any two *JourneyPattern* instances, at a particular stop or pair of stops, with default values for the connection activity.
  - A service may hold multiple connections.
  - The arrival stop of the inbound ‘feeder to’ journey, and the departure stop of the outbound ‘distributor from’ journey may be different *NaPTAN* stop points, i.e. require a transfer.
  - The mode of transfer (e.g. walk or otherwise) is indicated by a *TransferMode* property.

- A *VehicleJourneyInterchange* specifies the connection between two specific *VehicleJourney* (or *ConnectingVehicleJourney*) instances, at a *VehicleJourneyInterchange*. A vehicle journey connection can project onto an equivalent *JourneyPatternInterchange*, which constrains it to use the corresponding inbound feeder and outbound distributor journey pattern as in the reference, and the same stops specified by the *JourneyPatternInterchange*.

- A vehicle journey may have connections with more than one other vehicle journey.

- Specification of the connecting *VehicleJourney* may be done in either of two ways.
  - Using a normal *VehicleJourney*, with *VehicleJourneyTimingLinks* and full details
  - Using a *ConnectingVehicleJourney* – This allows a more lightweight statement of the connecting *VehicleJourney* without providing all of its details.

Note that inbound ‘feeder to’ and outbound ‘distributor from’ are relative roles; and a given service may serve as both feeder and distributor (i.e. passengers may exchange both ways between vehicles); in which case separate interchange instances can be declared for each direction.
3.11.1 Inheriting Interchange Values

Table 3-7 shows the various values that may be specified for the `JourneyPatternInterchange` and `VehicleJourneyInterchange` elements, and whether they are:

(i) Required ('R').
(ii) Optional but otherwise inherited from the previous level of discourse ('O').
(iii) Always Inherited. ('I').
For elements that are optional at all levels, a default value is identified to use if no explicit value is provided.

<table>
<thead>
<tr>
<th>Level</th>
<th>Property</th>
<th>Journey Pattern</th>
<th>Vehicle Journey</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interchange</td>
<td>InboundJourneyPatternRef</td>
<td>R</td>
<td>I</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>OutboundJourneyPatternRef</td>
<td>R</td>
<td>I</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>InboundStopUsageRef</td>
<td>R</td>
<td>I</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>OutboundStopUsageRef</td>
<td>R</td>
<td>I</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>InterchangeActivity</td>
<td>O</td>
<td>O</td>
<td>change</td>
</tr>
<tr>
<td></td>
<td>MinInterchangeTime</td>
<td>R</td>
<td>O</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>MaxInterchangeTime</td>
<td>O</td>
<td>O</td>
<td>Zero</td>
</tr>
<tr>
<td></td>
<td>InterchangeMode</td>
<td>O</td>
<td>I</td>
<td>walk</td>
</tr>
<tr>
<td></td>
<td>ValidityPeriod</td>
<td>O</td>
<td>O</td>
<td>service end date</td>
</tr>
<tr>
<td></td>
<td>StoppingArrangements</td>
<td>O</td>
<td>O</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternInterchangeRef</td>
<td>-</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>InboundVehicleJourneyRef</td>
<td>-</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>InboundStopPointRef</td>
<td>O</td>
<td>inherit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OutboundVehicleJourneyRef</td>
<td>-</td>
<td>R</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>OutboundStopPointRef</td>
<td>O</td>
<td>inherit</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3-10 – Interchange Properties and Defaults**

3.11.2 Interchange Schematic

*Figure 17* shows a schematic diagram of an interchange between two journeys. The inbound feeder journey arriving at stop ‘A’ from stop ‘X’ connects to a second distributor journey from stop ‘A’ onto stop ‘B’. The journey pattern interchange links the stop usages of the two journey patterns. The vehicle journey interchange links the two vehicle journeys.

![Interchanges Diagram](image)

**Figure 3-41 – Interchange Links**

3.11.3 Interchange Instance Example

As a pictorial example of a connection, *Figure 3-42* shows a UML instance diagram of the element instances for a connection between two vehicle journeys:
At the top, in yellow, can be seen a **Service** with two journey patterns, one inbound feeder to a **JourneyPatternInterchange**, and one outbound distributor from it. Each **JourneyPattern** has a single **JourneyPatternSection** containing a sequence of timing links; only the last **JourneyPatternTimingLink** of the inbound feeder journey pattern and the first **JourneyPatternTimingLink** of the outbound distributor journey pattern are shown, along with the **VehicleJourneyStopUsage** instance for each end of the link.

The **JourneyPatternInterchange** instance references both inbound feeder and outbound distributor journey patterns. It also references the destination **VehicleJourneyStopUsage** instance of the last timing link of the inbound feeder pattern, and the origin **VehicleJourneyStopUsage** of the last timing link of the outbound distributor pattern.

Below this, in orange, can be seen two corresponding inbound feeder and outbound distributor **VehicleJourney** instances. Again, only the last **VehicleJourneyTimingLink** of the inbound feeder vehicle journey and the first **VehicleJourneyTimingLink** of the outbound distributor vehicle journey are shown.

Each **VehicleJourneyTimingLink** individually projects onto the appropriate **JourneyPatternTimingLink** instance by an explicit reference.

Each vehicle journey has its own instance of a **VehicleJourneyInterchange**, which references both the inbound feeder and outbound distributor vehicle journey instances. It also references the **JourneyPatternInterchange** that connects the journey patterns upon which the vehicle journeys are based.
Interchange Instance Example

© Crown Copyright 2003-2009

Figure 3-42 – UML Instance Diagram of Example Interchange
3.12 Fare Stages

TransXChange supports the annotation of links with basic fare stage data for operational purposes. There are two different models commonly used for fare stages:

1. A **Stage Fare** model, where the fare stage is located on a boundary between two zones and is considered to be in both zones. In effect the fare stage is on the stop point, but only applies to journeys (i.e. sequences of links) where the other end of two subsequent links is in different zones.

2. A **Zonal** model, where the fare stage boundary lies between two stops, each within a distinct fare zone (**TariffZone**). The fare stage is in effect on the link between the stops. Only journeys going in the direction of the other zone and that cross the boundary will encounter the fare stage. One or more **TariffZones** can be specified for a stop using NaPTAN 2.5.

In the TransXChange model, fare stages are a property of timing link stop usage, so that both Stage Fare and Zonal models can be supported. Fare stage values can be specified at both the journey pattern and vehicle journey level of discourse as a successive property, that is one that carries onto succeeding links in the series until reset. The fare stage change occurs at the point of pick up, that is, at the originating end of the link, as shown in Figure 3-43, which shows examples of link sequences over a zone boundary for both fare models, with fare stage numbers and fare stage points marked. Whether a stop usage for a given link is a fare stage is properly determined by whether the **FareStageNumber** changes when traversing a sequence of timing links: the **FareStage** indicator can be used to store a statically computed determination of this property for convenience of implementation.

---

**Figure 3-43 – Fare Stages & Links**
3.13 Dead Runs

‘Dead run’ or positioning runs may be specified on vehicle journeys to describe how vehicles are placed in position to deliver a service, and also how they are retrieved after completing the service. Dead run positioning links are primarily of use for exchanging information for AVL systems, and are not needed for registration or publishing schedules. Dead runs can also be used to indicate short working. Figure 3-44 shows a UML structure diagram of the elements used to describe dead runs.

- A VehicleJourney may have an initial StartDeadRun and a final EndDeadRun.
- Each DeadRun consists of one or more PositioningLink instances.
  - Each PositioningLink runs between two position points, which may be specified as either a Location, a StopPoint, a LayoverPoint or a Garage.
  - A DeadRun may reference a VehicleJourneyTimingLink to indicate the point at which short working starts or stops.

Figure 3-44 – UML Diagram of Dead Run Model
3.13.1 Use of Dead Runs for Short Working

Dead runs may be used to indicate that a Vehicle Journey starts or ends at a particular point in a journey pattern, omitting all links & stops before or after the intercept point. See the Circular Route example for an illustration of both short and full workings of the same route.

3.13.2 Dead Run Hierarchy

Figure 3-45 shows the inheritance hierarchy for the **DeadRun** model elements.

![Figure 3-45 – UML Diagram of Dead Run Model Element Hierarchy](image)
3.14 The Registration Model

The statutory requirements of a bus registration are captured in TransXChange Registration by a submodel of Registration descriptive elements.

3.14.1 Introduction to the Registration Model

Figure 3-46 introduces the elements used to represent a Registration. These comprise:

- A TransXChange document can contain a Registration element:
  - A TransXChange Registration document must contain one Registration instance.
  - A TransXChange General document may contain one or more Registration instances.

- A single Service can be associated with each Registration.
  - A TransXChange Registration document Registration must contain a Service instance that references the Registration. It may have other Service definitions for connecting services.
  - A TransXChange General document may contain a Service instance.

- A Service has a RegisteredOperator, and may have additional AssociatedOperator instances. Operators may be instances of either LicensedOperator or Operator.
  - In a TransXChange Registration, the RegisteredOperator must be a LicensedOperator instance, with all details completed. (Note this constraint is enforced by an XML keyref).
  - In a TransXChange General document the RegisteredOperator may be an instance of either LicensedOperator or Operator.

- A Registration records the TrafficAreaNetwork and CirculatedAuthority instances.
  - Additional special details can be recorded for a ShortNoticeRegistration, including references to other services that the service replaces, or to which it connects. A short notice registration is an application to register, cancel or change a service made with less than the normally required 56 days’ period of notice.
  - The Registration can be annotated with SupportingDocument instances that identify related documents.

![Figure 3-46 – UML Diagram of Basic Registration Model](image-url)
3.14.2 Registration Model Details

Figure 3-47 elaborates Figure 3-46 to show the attributes of Registration elements.
**TransXChange Registration Model**

1. **Class TransXChange Registration Model**
   - **TransXChange Registration Model**
     - **RegistrationNumber**: 
       - «PK»
       - `registrationNumber`
     - **TrafficArea**: 
       - «PK»
       - `trafficArea`
     - **TanCode**: 
       - `tanCode`
     - **TrafficAreaName**: 
       - `trafficAreaName`
     - **VariationNumber**: 
       - `variationNumber`
     - **VosaRegistrationNumber**: 
       - `vosaRegistrationNumber`

2. **Registration Support**
   - **RegistrationVariantNature**: 
     - `registrationVariantNature`
   - **RegistrationWorkflowStatus**: 
     - `registrationWorkflowStatus`
   - **SubVariationNumber**: 
     - `subVariationNumber`
   - **RegistrationVariantNature**: 
     - `registrationVariantNature`

3. **Registration Information**
   - **ApplicationClassificationEnum**
   - **ContractingType**
     - «data type»
     - `contractingType`
   - **ContractingAuthority**
     - «data type»
     - `contractingAuthority`
   - **DisclaimerType**
     - «data type»
     - `disclaimerType`
   - **ContractedService**
     - «data type»
     - `contractedService`
   - **RoutingDescription**
     - «data type»
     - `routingDescription`
   - **ServiceRef**
     - «data type»
     - `serviceRef`

4. **Registration Details**
   - **DateOfRegistration**
   - **ExpirationDate**
   - **SubmissionDate**
   - **SubmissionAuthor**
   - **SupportingDocument**
   - **SupportingDocuments**
   - **SupportingDocuments**
   - **SupportingDocuments**

5. **Service Details**
   - **ServiceModel**: 
     - **Service**: 
       - «data type»
       - `service`
     - **ServiceRef**: 
       - «data type»
       - `serviceRef`

6. **Operational Details**
   - **OperationalRequirement**: 
     - «data type»
     - `operationalRequirement`
   - **OperationalSchedule**: 
     - «data type»
     - `operationalSchedule`
   - **OperationChange**: 
     - «data type»
     - `operationChange`
   - **OperationChange**: 
     - «data type»
     - `operationChange`
   - **OperatingTime**: 
     - «data type»
     - `operatingTime`

7. **Contracting Authority**
   - **ContractingAuthority**: 
     - «data type»
     - `contractingAuthority`
   - **ContractingType**: 
     - «data type»
     - `contractingType`

**Figure 3.47 – UML Diagram of TransXChange Registration**
3.14.3 Registration Model Workflow States

The **WorkflowStatus** attribute allows the status of a *Registration* to be represented for automated workflow processes. Figure 3-48 shows the allowed states and transitions.

![Figure 3-48 – State transitions for registration Workflow Status](image-url)

**Figure 3-48 – State transitions for registration Workflow Status**
3.14.4 Registration Model Hierarchy

Figure 3-49 shows the inheritance hierarchy for the Registration model elements.
3.14.5 Registration Model Support Types

Figure 3-55 shows the additional data types used in the Registration model elements.

```
<enumeration>
  ApplicationClassificationEnum
new
  changeable
  nonChangeable
  cancel
</enumeration>
<enumeration>
  RegistrationVariantNature
all
  footnotes
  journeyPattern
  other
  operational
  registration
  routeTrack
  serviceDetails
  textual
  timings
  vehicleJourneys
</enumeration>
<enumeration>
  TanCodeEnum
PF = Eastern Traffic Area
PB = North Eastern Traffic Area
PK = South East Metropolitan Traffic Area
PM = Scottish Traffic Area
PG = Welsh Traffic Area
PD = West Midlands Traffic Area
PH = Western Traffic Area
</enumeration>
<enumeration>
  TanCodeType
PF
PB
PK
PM
PG
PD
PH
</enumeration>
<enumeration>
  VosaRegistrationNumberType
  VariationNumberType
  VariationNumberTypeEnum
<enumeration>
  VariationNumber
type
</enumeration>
<enumeration>
  VariationNumber
  type
</enumeration>
<enumeration>
  VariationNumber
  type
</enumeration>
```

Figure 3-50 – UML Diagram of Registration Support types

3.14.6 Populating a Registration

Although it is legitimate for a Single Registered Service to have a number of journey pattern variants, the variation should be less than 50% of the primary journey pattern; i.e. *more than 50% of the mileage of the journeys should be in common, i.e. consist of vehicle journeys with timing links that visit or pass the same stops in the same order.*

3.14.7 Cancellation a Registration

When cancelling a registration it is not necessary to provide the entire schedule in electronic format. A restricted set of elements can be provided sufficient to identify the Registration and key particulars - the Registration, the Service & the Operator. A N example is provided.

3.14.8 Services and Registrations

Each registration file should contain all the journeys for a given service. Journeys which are operated by the same operator, use the same line number and follow the same journey pattern within the same operating period, should be in the same registration.
3.15 Operators

3.15.1 Introduction to the Operator Model

TransXChange includes a basic representation of an Operator to record who is making a Registration and who provides a Vehicle Journey. Figure 3-51 introduces the elements used to represent an Operator.

- LicensedOperator is a specialisation of Operator adds details about the licence or licences of the operator.
- An operator may have associated Garages and also DataRights (See later).
- AccessibilityBooking (V2.5) holds contact details for wheelchair bookings for the operator’s service. (See later).
- RegionOperatorRef provides a mapping of the Operator’s code for different regions if needed.

Figure 3-51 – UML Model of Operator

3.15.2 Operator Model Details

Figure 3-52 elaborates Figure 3-51 to show the attributes of Operator elements.
class TransXChange Operator Model

- **Name**: TransXChange Operator Model
- **Author**: nickk
- **Version**: 1.0
- **Created**: 17/02/2010 19:06:41
- **Updated**: 15/05/2013 12:28:18

### TransportModelModel

- **LicenceHolderNames**: LicenceHolderName [0..*]
- **LicenceExpiryDate**: date [0..1]
- **LicenceClassification**: LicenceClassificationType [0..1]

### VehicleModeEnum

- **underground**
- **trolleyBus**
- **tram**
- **rail**
- **metro**
- **ferry**
- **coach**
- **bus**

### NptgAdministrativeModel

- **RegionRef**: RegionCodeType* [0..1]
- **NationalOperatorRef**: NationalOperatorCodeType* [0..1]
- **OperatorRef**: OperatorCodeType* [0..1]

### AccessibilityBooking

- **AccessibilityBooking**

### OperatorSupport::DataOwnerEnum

- **EA = EastAnglia**
- **EM = East Midlands**
- **GL = Greater London**
- **NE = North East**
- **NW = North West**
- **SC = Scotland**
- **SE = South East**
- **SW = South West**
- **WA = Wales**
- **WM = West Midlands**
- **YO = Yorkshire**
- **XM = Combined MDV**

### OperatorSupport

- **OperatorCode**
- **PrimaryMode**
- **CommunityBusPermit**
- **SpecialRestricted**
- **StandardInternational**
- **StandardNational**
- **Restricted**
- **SpecialRestrictedCommunityBusPermit**

### LicenceStatusEnum

- **Valid**
- **Withdrawn**
- **Surrendered**
- **Refused**
- **ContinuationNotSought**
- **Revoked**

### DataRightsGroup

- **DataRights**

### OperatorLicence

- **OperatorNameOnLicence**
- **ReferenceName**
- **TradingName**
- **MultilingualString**
- **MultilingualString**
- **MultilingualString**
- **MultilingualString**
- **LicenseNumber**
- **OperatorLicenceNumberType** [0..1]
- **NationalOperatorCode**
- **NationalOperatorCodeType** [0..1]
- **LicenseStatus**
- **LicenseStatusEnum** [0..1]
- **LicenseExpiryDate**
- **date** [0..1]
- **LicenseHolderName**
- **LicenceHolderName** [0..1]
- **EbsrUser**
- **boolean** [0..1]
- **LicenseClassification**
- **LicenceClassificationType** [0..1]
- **ContactPerson**
- **normalizedString** [0..1]
- **PrivacyCode**
- **normalizedString** [0..1]
- **Address**
- **UkPostalAddress** [0..1]
- **Address**
- **UkPostalAddress** [0..1]
- **Location**
- **Location** [0..1]
- **PostCode**
- **PostCodeType** [0..1]
- **Line1**
- **normalizedString** [0..5]
- **TelephoneNumberType**
- **ContactTelephoneNumber** [0..1]
- **EnquiryTelephoneNumber** [0..1]
- **EmailAddress**
- **EmailType** [0..1]
- **ContractFaxNumber**
- **TelephoneNumberType** [0..1]

### OperatorModel

- **Operator**
- **VersionedObject**
- **AccessiblityBooking::AccessibilityBooking**

### Figure 3-52 – UML Diagram of TransXChange Operator Model
3.15.3 Operator / Accessibility Booking ((v2.5))

AccessibilityBooking (Figure 3-53) allows details about booking for assistance for wheelchair users to be recorded for an operator. Different values may be specified for different modes and regions.

The figure shows an UML diagram of the TransXChange Accessibility Booking Model. The diagram includes classes such as `OperatorModel`, `OperatorSupport`, `AccessibilityBooking`, and `BookingArrangements`. It illustrates the relationships and properties of these classes, such as booking methods, accessibility arrangements, and operator support.

Figure 3-53 – UML Diagram of TransXChange Accessibility Booking Model
3.15.4 Operator Model Hierarchy

Figure 3-54 shows the inheritance hierarchy for the Operator model elements.

Figure 3-54 – UML Diagram of Operator Element Hierarchy
3.15.5 Operator Model Support Types

Figure 3-55 shows the additional data types used in the Operator model elements.

3.16 Further Modelling Topics

3.16.1 Direction: Handling Inbound and Outbound Schedules.

A Service may contain both inbound and outbound journeys, comprising in effect two distinct timetables for the two directions. Normally completely separate routes will be specified for each direction, because there are typically separate NaPTAN points for bus stop pairs each side of the road; routes will therefore be following a different sequence of stops along slightly different road sections. However, there are scenarios where the route (and associated sequence of stops) in one direction is an exact reversal of the route (and associated sequence of stops) in the opposite direction. In this case it is possible to share the route definitions for both directions of a service, as follows (Figure 3-56).

1. Each Route contains one or more route sections, each containing a sequence of route links. Each route link is flagged as Outbound, Inbound, Clockwise or Anticlockwise. All the links within a route section must be in the same direction.

2. At least one journey pattern is specified for each direction of the Route. The journey pattern sections contain journey pattern timing links in the order of traversal, each of which can specify a direction (if a direction is not specified the direction will be assumed to be the same as that of any route link which the timing link references).
   - If the direction of a journey pattern timing link is the same as that of the route link which it references, then the stops referenced in the “from” and “to” stop usages of the timing link will be the same as for the route link, and the timing links will appear in the same order as the route links.
o For example, if (a) Route Link ‘RL_1’ goes from ‘A’ to ‘B’ with a direction of ‘outbound’, and (b) Route Link ‘RL_2’ goes from ‘B’ to ‘C’, also with a direction of ‘outbound’, then the outbound journey pattern would have two outbound journey pattern timing links: (i) ‘JTL_1’ which references ‘RL_1’ with a direction of ‘outbound’, and also runs from ‘A’ to ‘B’, followed by (ii) journey pattern timing link ‘JTL_2’, which references ‘RL_2, and goes from ‘B’ to ‘C’. Note that in this discussion; ‘A’, ‘B’, etc refer to stop pairs: in actuality, the inbound and outbound stops are likely to be distinct stops of a pair either side of the road. So actually the NaPTAN stops of inbound and outbound routes and journey pattern will be quite distinct.

- If the direction of the of a journey pattern timing link is the opposite to that of the route link which it references, then both the link order, and the stops referenced in the from and to stop usages will be reversed.

  o For example, if (a) Route Link ‘RL_1’ goes from ‘A’ to ‘B’ with a direction of ‘outbound’, and (b) Route Link ‘RL_2’ goes from ‘B’ to ‘C’, also with a direction of ‘outbound’, then the inbound journey pattern would have two inbound journey pattern timing links: (i) journey pattern timing link ‘JTL_X1, which references ‘RL_2’ but which runs from ‘C’ to ‘B’, and (ii) journey pattern timing link ‘JTL_X2,’ which references ‘RL_1’ but which runs from ‘B’ to ‘A’.

3. Each vehicle journey follows the same direction as the journey pattern that it references.

4. The Service may be given an overall Direction: this may be one of Inbound, Outbound, InboundAndOutbound, Clockwise, Anticlockwise, or Circular.
The TransXChange Publisher will sort the vehicle journeys of a service into distinct outbound and inbound groups, and create a separate matrix for each direction.

3.16.2 Modelling Complex Routes

The TransXChange model can be used to represent complex services, for example:

- Services with topologically complex routes.
- Services with complex temporal operational patterns.

3.16.2.1 Services with Topologically Complex Routes

The TransXChange model can be used to represent complex patterns of service:

1. **Repeated stop routes.** Circular (Figure 3-57), Lollipop (Figure 3-58), and Cloverleaf (Figure 3-59) routes involve visiting the same stop more than once within a single vehicle journey. In the TransXChange model, each link has a separate identity in both the route, journey pattern and vehicle journey link sequences, so it is possible to distinguish the separate link traversals and occurrences of a stop in a journey, and so to compose complex routes, and also to
project unambiguously the links of such routes between the route, journey pattern and vehicle journey level of discourse. (In TransXChange 1.2 this was not always possible).

Other features helpful in representing complex routes are:

- **Dynamic destination displays**, so that bus headings can change over the course of the route.
- Reusable route and journey pattern **sections**, so that definitions of sections of the route and/or journey pattern may be shared between different journeys. See ‘Modelling Services Efficiently’ below.
- **Stop Sequence numbers** – so that the presentation of a route in a matrix can be exactly controlled. See ‘Presenting Schedules in Timetables’ below.

2. **Multiple route variants.** Complex services may be composed of multiple route and journey pattern variations, either involving covering different branches of the physical network, or traversing subsets of the full stop sequence, or both.

- **Line** elements can be used to separate the **modelling** of the network topology as routes and journey patterns, from the **labelling** of the network services with public identifiers on vehicle journeys – which is done using the **Line / LineName** element. Thus several different route variants may all be grouped under the same line name.
- **RouteSection** elements can be used to model reusable subsections and branches of the route network, and **JourneyPatternSection** elements can be used to annotate this substructure with timing values, allowing for the representation and reuse of the route substructure.

3. **Connecting routes.** The connections between routes services may be described using **JourneyPatternInterchange** and **VehicleJourneyInterchange** elements.

---

**Figure 3-57 – Topology: Circular Route**

**Figure 3-58 – Topology: Lollipop Route**
Figure 3-59 – Topology: Cloverleaf Route
3.16.2.2 Services with Complex Temporal Operational Patterns

The TransXChange model can be used to represent complex operational times. This is discussed in detail under 'Modelling Operational Days' below. All of the following mechanisms are available:

- **Regular day types**: Days of week, Day Combinations, Weeks of month.
- **Special day types**: Bank holidays.
- **Date ranges**: Operating Period, Validity Periods, and Exceptions.
- **Time bands**: Time bands of operation of flexible services.
- **Frequency based**: Interval or minute patterns of operation of frequency based services.
- **Serviced Organisations**: Term-times, holidays or other events of specified organisations.
3.16.3 Modelling Services Efficiently

TransXChange supports an extensive reuse of service and journey description elements so that an efficient encoding of journeys can be achieved. In particular:

- Existing data reference systems can be used; NaPTAN stop and stop area definitions, NPTG localities.
- Elements describing the network topology and other shared infrastructure entities can be declared once, and then be reused by a simple reference. Notably:
  - Topographical elements: StopPoint, StopArea.
  - Network layer elements: Route, RouteSection.
  - Supply elements: JourneyPatternSection, and JourneyPattern.
- Link Sequences – the timing link sequences of journey patterns and vehicle journeys – may be reused in several different ways:
  a. A JourneyPatternSection may be used in many different JourneyPattern instances.
  b. A given JourneyPattern may be referenced in many different VehicleJourney instances, and its values inherited. Only those individual vehicle journey timing links whose properties are different from the corresponding timing links of the underlying journey pattern need be specified.
  c. A VehicleJourney may specify that particular stops of a referenced JourneyPattern are omitted, allowing for "express" journeys constrained to a basic journey, and for short working.
  d. A VehicleJourney may reference another VehicleJourney to share the timing links of that specific journey.
  e. The Frequency element of VehicleJourney may be used to indicate that the same vehicle journey is repeated to the same pattern many times at regular intervals.
- Operational day types and dates may be reused.
  - The OperatingProfile specified for a Service can be shared by all the service’s vehicle journeys. Individual JourneyPattern and VehicleJourney instances need only state their specific differences from the base values.
- Properties of successive links need only be specified when they change:
  - The successive properties of links, such as fare stages and dynamic headings, do not have to be repeated on every link, but only need to be specified when they change from the preceding link.

It remains up to the implementer to decide the degree of reuse that she wishes to achieve. A verbose implementation may, if it wishes, re-declare stops and create separate route, route section, route links, journey pattern, journey pattern section, journey pattern links, operation profile, and special operation profile instances for every single vehicle journey. However it should be noted that a verbose
implementation (a) wastes space (b) may fail to exchange structural information about the underlying schedule.

3.16.3.1 Overall Reuse of Elements

*Figure 3-60* shows some of the ways that elements may be reused at different levels of discourse.

---

### Reuse in TXC 2.0

![Reuse Diagram](image)

**Figure 3-60 – Reuse of Elements**

3.16.3.2 Inefficiencies in TransXChange

Although inheritance, default values and reuse can be used to optimise document content, *TransXChange* is not a fully optimised representation, and has a number of data redundancies in its representation. In particular:

- Start and end stop usages are repeated on every successive link in a journey pattern and vehicle journey, including the stop point reference on the journey pattern usage.
- Start and end stop points are repeated on both route links and journey pattern links.
3.16.3.3 Use of Sections

Route sections allow implementers to reuse a sequence of stops and links in more than one route. Journey pattern sections allow the corresponding sequence of timing links to be reused. Figure 3-61 shows an example of a service comprising three named lines (Line 54 Line 54A and Line 12). The lines are made up of four routes, containing five sections that group the eleven different links of the network. Two of the sections (S1 & S2) are reused in two different routes.

- **Line 54**
  - R1 = S1(L1, L2) + S2(L3 + L4 + L5 + L6)
  - R2 = S1(L1, L2) + S4(L3 + N1 + N2 + L6)

- **Line 54A**
  - R3 = S3(M1) + S2(L3 + L4 + L5 + L6)

- **Line 12**
  - R4 = S3(M1) + S5(R1, R2)

![Figure 3-61 – Example of Sections](image-url)
3.16.4 Presenting Schedules in Timetables

TransXChange is primarily concerned with the representation of schedule data for exchange between different computer systems, and is not intended to address all the additional requirements for presenting schedules as published representations for the public. However, it is possible to transform a TransXChange schedule into a matrix timetable format automatically, adopting a specific order for showing the stops. Rendering the journeys in a tabular format is valuable because it allows a TransXChange document to be validated by human inspection against the originating and published formats.

The TransXChange Publisher provides an example of a matrix rendering, which follows a conventional mapping:

- **VehicleJourney** instances generally correspond to columns.
  - Each VehicleJourney instance can have a VehicleJourney / Note associated with it.
  - Each VehicleJourney can have an OperatingProfile to specify operational time information specific to it in a quantitative structure.
  - If a Frequency is specified, one or more additional columns may be interpolated to indicate the repeating journeys.
  - If a Frequency and the same EndTime is specified for more than one journey, one or more journey columns may be merged to create a single frequency group. See 3.18.8.4.
  - VehicleJourney are ordered as columns across the matrix in the same order as they are declared in the document. Normally they should be sorted into time order.
- **JourneyPatternTimingLink / VehicleJourneyStopUsage** instances generally correspond to each individual row.
- **VehicleJourneyTimingLink / VehicleJourneyStopUsage** instances generally correspond to cells for each individual row.
  - Each From / VehicleJourneyStopUsage corresponds to a departure. Normally the departure is shown for all stops of the route except the last.
  - Each To / VehicleJourneyStopUsage corresponds to an arrival. Normally arrivals are only shown for the last stop.
  - If the arrival and departure time is different at a stop, two separate rows for arrival and departure will be shown.
  - The stop rows will be ordered down the page in the same order that VehicleJourneyStopUsage instances appear in the Journey pattern of the Vehicle Journey (unless overridden by a SequenceNumber).
  - To collate different journeys that follow different journey patterns into a single matrix, the publisher compiles a list of all the stops of all the journey patterns, in order of use. The stops of each vehicle journey are aligned against this list, leaving an empty cell for any stop that is not visited by a particular journey. Thus for example, if two journey patterns A-B-F and A-C-D-F are combined as list, these will be collated as A-B-C-D-F, resulting in column entries A-B-()-()-F for the first and A-()-C-D-F for the second.
  - Within this overall ordering, where there are stops that are specific to particular journey patterns these will be ordered according to the passing time. Thus for example, consider two vehicle journeys using separate patterns A-B-D and A-C-D, which have passing times at stop A(t1) - B(t2) - D(t3) and A(t2) - C(t3) - D(t4) where t indicates the relative time. If these are combined as list, these will be ordered A-C-B-D, rather than say A-B-C-D because stop C is visited earlier (t3) than stop B (t4).
  - As a further refinement to the overall ordering, the publisher uses the grouping of stops given by the JourneyPatternSection to sequentially order a series of stops in succession that are a route variants used only by certain journeys, rather than the strict relative time. Thus for example, if two journey patterns A-[B-C-D]-F and A-[P-Q-R]-F are combined as list these will be ordered as A-B-C-D-P-Q-R-F, rather than say as A-B-P-C-Q-D-R-F, regardless of the relative passing times at BCD and PQR. This results in more readable columns entries of A-B-C-...
A `SequenceNumber` attribute can be specified on individual `JourneyPatternStopUsage` instances to suggest a preferred sort order of stops for presentation. When listing the stops as rows in a matrix, the explicit number overrides the default traversal sequence that will be otherwise assumed for publication. Note that each vehicle journey is still traversed by a bus in the actual order of its links regardless of any `SequenceNumber` instances.

3.16.4.1 Using a Sequence Number

The `SequenceNumber` attribute on individual `JourneyPatternStopUsage` instances allows you to control the ordering of stops in tabular presentations.

1. Every stop usage of a journey pattern timing link can be allocated a sequence number (i.e. both the departure-from-stop end, and arrival-at-stop end).
2. Either all of the stop usages ('explicit numbering'), or none of the stop usages ('implicit numbering') of a complete journey pattern should be numbered. (N.B. If some are numbered and some are not, indeterminate effects may occur in applications that make use of the `SequenceNumber`.)
   - For implicit stop numbering, each stop usage is consequentially numbered in order of traversal of the timing links of the route and journey pattern. (Note that journey pattern timing links must in any case visit the same stops in the same order as the route links of any route that the journey pattern references).
   - For explicit stop numbering, each stop usage is consequentially numbered in the implementer’s preferred presentation order. The actual traversal of the journey timing links for the computation of passing times still follows the sequence of the links of the journey pattern, even if the stops are sequenced in a different order by means of a `SequenceNumber`. 

D-(·)-(·)-(·)-F and A-(·)-(·)-(·)-P-Q-R-F, rather than say A-B-(·)-C-(·)-D-(·)-F and A-(·)-P-(·)-Q-(·)-R-F.
For example, the TransXChange Publisher uses the `SequenceNumber` as follows:

1. The publisher builds a matrix by creating a line for each vehicle journey stop usage (i.e. arrival and departure, of each vehicle journey in the service), and sorting them all into stop sequence order.
   - If there are several different underlying journey patterns (i.e. routes) making up the overall service, giving rise to overlapping (or even completely disjoint) sets of stops, the publisher takes the combined set of all stop usages; if the same stop usage appears with the same sequence number in multiple journey patterns, then it is shown as the same stop row.
2. For each column, i.e. vehicle journey, the stop passing times for each stop are computed in the order of traversal of the timing links; times are only shown in cells for the stops that are visited.
3. If the arrival and departure usages for the same stop appear on consecutively lines on the matrix, they can be shown as a single line, showing just the departure time.

3.16.4.2 Example of a Timetable using StopSequence

Figure 3-62 shows a service with two alternate routes (R1 & R2) over six stops (labelled alphabetically ‘A’ to ‘F’) and which are labelled; line ‘1C’, which runs ‘A-B-C-E-F’, and line ‘1D’, which runs ‘A-B-D-E-F’.

![Sequenced Timetable ("Matt’s Eye")](image-url)
In the published timetable for the service, the preferred presentation might be to show the two journeys aligned on all similar stops – see Table 3-11.

<table>
<thead>
<tr>
<th>Journey</th>
<th>SequenceNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1C 10:00</td>
</tr>
<tr>
<td>B</td>
<td>1D 10:05</td>
</tr>
<tr>
<td>C</td>
<td>1D 10:10</td>
</tr>
<tr>
<td>D</td>
<td>1D 11:12</td>
</tr>
<tr>
<td>E</td>
<td>1D 10:15</td>
</tr>
<tr>
<td>F</td>
<td>1D 10:20</td>
</tr>
</tbody>
</table>

Table 3-11 – Example: Eye Timetable with Explicit Stop Sequencing

To specify the above presentation we might do the following:

- Break the two routes down into four sections containing route links as follows:
  - \( R_1 = RS_1(RL_1) + RS_2(RL_2, RL_3) + RS_4(RL_4) \)
  - \( R_2 = RS_1(RL_1) + RS_3(RL_5, RL_6) + RS_4(RL_4) \)
- Define a journey pattern, JP1, over route \( R_1 \), specifying a preferred stop sequence \( n \) for each end of each timing link:
  - \( JP_1 = \begin{cases} JPTL_1[5mn, \text{from}:1, \text{to}:3] \\ JPTL_2[5mn, \text{from}:2, \text{to}:3] \\ JPTL_3[5mn, \text{from}:3, \text{to}:5] \end{cases} \)
- Define a journey pattern, JP2, over route \( R_2 \): also specifying a preferred stop sequence:
  - \( JP_2 = \begin{cases} JPTL_4[5mn, \text{from}:5, \text{to}:6] \end{cases} \)

As a comparison, Figure 3-67 shows the default ordering that would be used by the TransXChange Publisher if no sequence guidance was given. Note this is a difference of presentation, not representation – in the underlying TransXChange document, each individual vehicle journey is still correctly ordered as to its sequence of visiting the stops by virtue of its journey pattern. If wait times had been specified, then arrival and departure would be distinct.

<table>
<thead>
<tr>
<th>Stop</th>
<th>1C</th>
<th>1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10:00</td>
<td>10:00</td>
</tr>
<tr>
<td>B</td>
<td>10:05</td>
<td>10:05</td>
</tr>
<tr>
<td>C</td>
<td>10:10</td>
<td>-</td>
</tr>
<tr>
<td>E</td>
<td>10:15</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>10:17</td>
</tr>
<tr>
<td>E</td>
<td>10:15</td>
<td>11:17</td>
</tr>
<tr>
<td>F</td>
<td>10:20</td>
<td>11:20</td>
</tr>
</tbody>
</table>

Table 3-12 – Example: Eye Timetable with Implicit Stop Sequencing
3.16.5 Grouping Journeys within a Timetable

TransXChange includes mechanisms for specifying how journeys should be grouped within a service. This grouping can be used by the publisher and other tools to present journeys grouped by day type or other criteria. A text description can be associated with each of the groupings. In some circumstances, for example for large but sparsely populated or complex timetables moving one or more exceptional journeys to a separate matrix can greatly reduce the overall size of the matrix.

By Default six ‘built-in’ Journey Groupings are assumed for each Service:
- **Outbound**: Monday to Friday, Saturday and Sunday.
- **Inbound**: Monday to Friday, Saturday and Sunday.

These built-in journey groupings do not have to be explicitly specified unless you wish to associate a description or note with them.

Additional **CustomJourneyGroupings** may be added (+TXC v2.4). Each **CustomJourneyGrouping** must explicitly specify the **VehicleJourney** instance to be included. Any custom journeys will be omitted from the built-in Journeys.

3.16.5.1 Journey Grouping Model Overview

Figure 3-67 gives an overview of the **JourneyGrouping** elements.

---

**Figure 3-67 – UML Diagram of Service Journey Grouping elements - Introduction**
3.16.5.2 JourneyGrouping Elements

Figure 3-64 shows the JourneyGrouping elements in more detail.

![UML Diagram of Service Journey Grouping elements – Details](image)

3.16.5.3 JourneyGrouping Model Hierarchy

Figure 3-65 shows the inheritance hierarchy for the JourneyGrouping model elements.
3.16.6 Specifying services that run before or after midnight

The operating profile of a journey states the day or days of the week on which it runs, for example, ‘Monday to Friday’, ‘Monday to Saturday’, Sunday, etc. This is normally the day in which the departure time of the journey falls.

Sometimes however OPERATING DAY may be different, for example services for a given operating day might run from 2 am to 2 am. For example:

1. **Late night service.** A service that starts in the early morning after midnight but that is part of the previous evening’s service (i.e. operating day). For example, a service that runs at 01:10 Tuesday to Saturday nights as part of a Monday to Friday service.

2. **Early morning service.** A service that starts late at night before midnight but that is part of the next day’s service. For example, a service that runs at 23:55 Sunday to Thursday nights as part of a Monday to Friday service.

This can be achieved by means of the **DayShift**. The day shift the times of the journey forward or back by twenty four hours. :

- A negative **DayShift** indicates a journey time is in the previous day and should be shown with a footnote ‘**Previous day’.
- A positive **DayShift** causes indicates a journey time is in the next day and should be shown with a footnote ‘**Next day’.

Journeys should be placed in a TransXChange document in their true temporal order, that is (i) negatively shifted journeys in departure time order, (ii) unshifted journeys in departure time order and (iii) positively shifted journeys in departure time order.
3.17 DayShift Examples

3.17.1 Late night Weekday service – Positive DayShift

Table 3-13 shows an example of a Monday to Friday service where there is a post-midnight early morning journey (#3) that actually takes place on Tuesday Saturdays at 00:30.

3.17.1.1 Actual Journeys

<table>
<thead>
<tr>
<th></th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DayOfWeek</td>
<td>MF</td>
<td>MF</td>
<td>MF</td>
</tr>
<tr>
<td>Actual days</td>
<td>MTWTF--</td>
<td>MTWTF--</td>
<td>TWTFS -DayShift +1</td>
</tr>
<tr>
<td>A</td>
<td>20:30</td>
<td>21:30</td>
<td>00:30</td>
</tr>
<tr>
<td>B</td>
<td>20:40</td>
<td>21:40</td>
<td>00:40</td>
</tr>
<tr>
<td>C</td>
<td>21:10</td>
<td>22:10</td>
<td>01:10</td>
</tr>
</tbody>
</table>

Table 3-13 – Positive DayShift example

3.17.1.2 As Shown in Beds with Positive DayShift

Journey #3 is given a positive DayShift of ‘+1’, which has the following effect:
- Indicates to journey planners etc that he service is actually in the next day
- Causes it be flagged with a foot note indicating it runs on the next day.

<table>
<thead>
<tr>
<th></th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>20:30</td>
<td>21:30</td>
<td>00:30** Next Day</td>
</tr>
<tr>
<td>B</td>
<td>20:40</td>
<td>21:40</td>
<td>00:40** Next Day</td>
</tr>
<tr>
<td>C</td>
<td>21:10</td>
<td>22:10</td>
<td>01:10** Next Day</td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td></td>
<td>Tuesday to Saturday</td>
</tr>
</tbody>
</table>

Table 3-14 – Positive DayShift example - effect
3.17.2 Early morning Weekday Service – Negative DayShift

Table 3-15 shows an example of a *Monday to Friday* service where there is a pre midnight late evening journey (¶1) that actually takes place on *Sunday to Thursdays* at 23:50.

### 3.17.2.1 Actual Journeys

<table>
<thead>
<tr>
<th>Stop</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DayOfWeek</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MF</td>
<td>MF</td>
<td>MF</td>
</tr>
<tr>
<td>Actual days</td>
<td>MTWT--S</td>
<td>MTWTF--</td>
<td>MTWTF--</td>
</tr>
<tr>
<td>DayShift - 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>23:30</td>
<td>10:30</td>
<td>21:30</td>
</tr>
<tr>
<td>B</td>
<td>23:40</td>
<td>10:40</td>
<td>21:40</td>
</tr>
<tr>
<td>C</td>
<td>00:10</td>
<td>11:10</td>
<td>22:10</td>
</tr>
</tbody>
</table>

**Table 3-15 – Negative DayShift example:**

#### 3.17.2.2 As shown in Beds with Positive DayShift

Journey #3 is given a negative *DayShift* of ‘-1’, which has the following effect:

- Indicates to journey planners etc that the service is actually in the previous day
- Causes it to be flagged *with a footnote* indicating it runs on the previous day.

<table>
<thead>
<tr>
<th>Stop</th>
<th>#3</th>
<th>#1</th>
<th>#2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DayOfWeek</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MF</td>
<td>MF</td>
<td>MF</td>
</tr>
<tr>
<td>Actual days</td>
<td>MTWT--S</td>
<td>MTWTF--</td>
<td>MTWTF--</td>
</tr>
<tr>
<td>DayShift - 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>23:30*Previous day</td>
<td>20:30</td>
<td>21:30</td>
</tr>
<tr>
<td>B</td>
<td>23:40*Previous day</td>
<td>20:40</td>
<td>21:40</td>
</tr>
<tr>
<td>C</td>
<td>00:10</td>
<td>20:50</td>
<td>21:50</td>
</tr>
<tr>
<td>Note</td>
<td>Sunday to Friday</td>
<td></td>
<td>MF</td>
</tr>
</tbody>
</table>

**Table 3-16 – Positive DayShift example: Monday to Friday Bed**

#### 3.17.3 Associating Operational Data with a Timetable

*TransXChange* provides several means of associating different types of operational data with the elements of a timetable (See Figure 3-66). For example,

- *JourneyPatterns & VehicleJourneys* may be associated with an *Operational* element that specifies a *Block, VehicleType* or *TicketMachine* for a journey.
- *The VehicleType* may include basic accessibility data such as whether the vehicle is considered accessible, and some further details of the *VehicleEquipment*.
- *A TimingLink* may specify a *DutyCrew* for a link.
3.17.3.1 Operational Data Model Introduction

Figure 3-66 – UML Diagram of Operational data elements: Introduction

3.17.3.2 Operational Data Model Details

Figure 3-67 shows the details of the Operational model.

Figure 3-67 – UML Diagram of Operational data elements: Details
3.17.4 Operational Model Hierarchy

Figure 3-68 shows the inheritance hierarchy for the Operational model elements.

![UML Diagram of Operational Element Hierarchy]

3.17.4.1 Operational Data Model Support Types

Figure 3-69 shows the additional data types used by the Operational model elements.

![UML Diagram of Operational Support types]

3.17.5 Vehicle Equipment Model (+TXC v2.4)

Figure 3-70 shows the vehicle equipment data types that may be used to specify the accessibility facilities of a vehicle. These are based on the IFOPT/NeTEx model.
class XSD TransxChange (NeTEx) Vehicle Passenger Equipment Model

PassengerEquipment
- PassengerInfo: PassengerInfoEnum [0..*]
- AccessibilityInfo: AccessibilityInfoEnum [0..*]

«PK»
- Id: VehicleInfoEquipmentIdType

PassengerEquipment
- WheelchairVehicleEquipment
  - NumberOfWheelchairAreas: integer [0..1]
  - WidthOfAccessArea: Length [0..1]
  - HeightOfAccessArea: Length [0..1]
  - WheelchairTurningCircle: Length [0..1]

«PK»
- Id: VehicleWheelchairIdType

«enumeration»
- VehicleEquipmentValues:: AccessibilityInfoEnum
  - audioInformation
  - audioForHearingImpaired
  - visualDisplays
  - displaysForVisuallyImpaired
  - tactilePlatformEdges
  - tactileGuidingStrips
  - largePrintTimetables
  - other

«enumeration»
- VehicleEquipmentValues:: PassengerInfoFacilityEnum
  - nextStopIndicator
  - stopAnnouncements
  - passengerInformationFacility
  - other

«view»
- VehicleTypeModel::VehicleType
  - VehicleType: VehicleTypeCodeType [0..1]
  - Description: MultilingualString [0..1]
  - ServiceFacilitySet: ServiceFacilitySet [0..1]
  - Id: VehicleInfoEquipmentIdType

PassengerEquipment
- AccessVehicleEquipment
  - LowFloor: boolean [0..1]
  - Hoist: boolean [0..1]
  - HoistOperatingRadius: LengthType
  - Ramp: boolean [0..1]
  - RampBearingCapacity: Weight [0..1]
  - NumberOfSteps: integer [0..1]
  - BoardingHeight: LengthType [0..1]
  - GapToPlatform: LengthType [0..1]
  - WidthOfAccessArea: LengthType [0..1]
  - HeightOfAccessArea: LengthType [0..1]
  - AutomaticDoors: boolean [0..1]
  - SuitableFor: MobilityNeed [0..*]
  - AssistanceNeeded: AssistanceNeededEnum [0..1]
  - AssistanceBoardingLocation: AssistedBoardingLocationEnum [0..1]
  - GuideDogsAllowed: boolean [0..1]

«enumeration»
- VehicleEquipmentValues:: AssistedBoardingLocationEnum
  - boardAtAnyDoor
  - boardOnlyAtSpecifiedPositions
  - unknown

«enumeration»
- VehicleEquipmentValues:: AssistanceNeededEnum
  - levelAccess
  - rampRequired
  - hoistRequired
  - assistanceRequired

«enumeration»
- AccessibilityModelValues:: MobilityNeedEnum
  - wheelchair
  - assistedWheelchair
  - motorizedWheelchair
  - mobilityScooter
  - normalMobility
  - unknown
  - roadMobilityScooter

Figure 3-70 – UML Diagram of VehicleEquipment Types
3.17.6 Assistance Service Model (+TXC v2.5)

Figure 3-71 shows the **AssistanceService**, a type of local service “Equipment” that may be used to specify details and conditions about assistance available for a **VehicleJourney**. These are based on the NeTEx model.

![UML Diagram of Assistance Service Types](image)

**Figure 3-71 – UML Diagram of Assistance Service Types**

3.17.7 ServiceFacilitySet Model (+TXC v2.5)

A **ServiceFacilitySet**, Figure 3-71, can be used to list facilities available on a service or specific **JourneyPattern** or **VehicleJourney**. The facilities are just list of named properties – any detailed attributes of a specific facility are described by a corresponding **Equipment** element. The values supported are a subset of those found in the NeTEx model.

Each facility value specified at the Service level apply to all journeys unless overridden on a specific **JourneyPattern** or **VehicleJourney**.
Figure 3-72 – UML Diagram of ServiceFacilitySet Introduction

3.17.7.1 ServiceFacilitySet Data Model Details

Figure 3-73 shows the details of the ServiceFacilitySet model.
3.17.7.2 ServiceFacilitySet Data Model – Common values

Figure 3-74 shows some the Facility values that may be specified for a ServiceFacilitySet. These include FareClass, CateringFacility, NuisanceFacility, PassengerCommsFacility, PassengerInformationFacility, SafetyFacility, and SanitaryFacility.

Figure 3-74 – UML Diagram of ServiceFacilitySet: Common values

3.17.7.3 ServiceFacilitySet Data Model – Specific values

Figure 3-75 shows three further lists of values that may be specified for a ServiceFacilitySet: AccommodationFacility, LuggageCarriageFacility and ServiceReservationFacility.
class XSD TXC NeTEx RC Service Facility Values

{Constrained to those facilities relevant to SERVICE}

FacilitySet

ServiceFacilitySet

«enumeration»

AccommodationFacilityEnum

seating

doubleSleep

singleSleep

specialSleep

couchette

recliningSeat

babyCompartment

familyCompartment

panoramaCoach

pullmanCoach

standing

«enumeration»

LuggageCarriageEnum

noBaggageStorage

luggageRacks

extraLargeLuggageRacks

baggageVan

cyclesAllowed

cyclesAllowedInVan

cyclesAllowedInCarriage

cyclesAllowedWithReservation

vehicleTransport

noCycles

«ImplementAsListOfEn...»

FacilityList

There is a separate named enumerated list for each type of facility value

«enumeration»

LuggageCarriageFacilityList

«enumeration»

ServiceReservationFacilityList

«ImplementAsListOfEn...»

AccommodationFacilityList

« enumerati...»

ServiceReservationSubsetFacilityEnum.7037

seatReservationsCompulsory

bicycleReservationsCompulsory

seatReservationsRecommended

seatReservationsPossible

wheelchairOnlyReservations

reservationsCompulsoryForGroups

reservationsPossibleForCertainClasses

noReservationsPossible

Figure 3-75 – UML Diagram of ServiceFacilitySet: Specific values
3.17.8 Service Descriptions

The TXC schema allows descriptive elements to be declared that describe where a service runs:

(a) Each Service may have a Description. This is non-directional. In a registration there is only a single Service. In a general document there may be several Services.
   - For each Service an Origin place name, Destination place name and a list of Via place names can also be specified, these can be used to generate text descriptions for the Service.
   - The Service / Description is published in the Service particulars.

(b) Each Line of a Service may have an OutboundDescription and an InboundDescription (added in TXC v2.4).
   - Each description can have an Origin place name, Destination place name and a list of Vias place names can also be specified. These can be used to generate directional text descriptions for the Service. There might be more than one line for a service, and so multiple line descriptions.

(c) Each Route may have a Description. There may be many Routes, each with one or more journey patterns and hence one or more vehicle journeys associated with it. It is published in a list of route names as part of the service particulars. Separate Routes can (and often are) be used each direction, so it can be used to get a separate description for each direction. However the Route Description cannot be related to specific journeys or matrix beds, since a matrix bed may reference more than one route.
   - The Route / Description is published in the Service particulars.

(d) Each JourneyPattern may have a Description. This is directional. There may be many journey patterns, each with one or more vehicle journeys in a service. Since there may be more than one journey pattern associated with a single matrix bed, one cannot necessarily determine which journey pattern (and hence description) to use for the bed.
   - The value is not currently published.

(e) Each VehicleJourney may have a Description. This is directional – It applies to the direction of the individual journey. There will be many vehicle journeys associated with a given matrix bed.
   - The value is not currently published.

(f) A Description element on each JourneyGrouping. (+TXC v2.4) There will be many vehicle journeys associated with a given JourneyGrouping.
   - The value is published as the title for each matrix bed. (TXC 2.4)

3.18 Modelling Operational Days

TransXChange has rich (and complex!) capabilities for specifying the operational days and times of a bus service for both regular running, and for exceptional days. We introduce these capabilities here. For further details, see the descriptions of individual schema elements. For an overall summary of how to combine date conditions, see also Section 13 ‘Integrity Rules’.

3.18.1 Specifying When the Service Operates – Summary

The OperatingProfile specifies when a bus service runs as a set of day properties, including both the types of days (e.g. Monday to Friday) on which the service normally runs; and what happens on special days such as Bank Holidays, and can also describe any exceptional periods of operation.

- An overall OperatingPeriod can also be specified at the service level. This can be open ended.
- Default OperatingProfile values can be specified at the Service level, and be overridden at both the JourneyPattern level and on individual VehicleJourney instances.
Validity periods can also be specified for the operation of *JourneyPatternInterchange* instances, constraining the availability of interchanges between specified *VehicleJourney* instances.

Operational times may also be specified in terms of the working days of a *ServicedOrganisation* – see later below.

An operational *Calendar* can be used to relate the Day Types to specific dates in a given year.

*Figure 3-76*, in UML class diagram notation, gives a high-level view of the main elements and relationships concerned with operational days.

![UML Diagram Overview of Operational Times](image-url)
3.18.2 Regular Operation – OperatingProfile

The OperatingProfile / Normal group specifies the normal operating day types of a service. It can be made up of three elements, as shown in the UML structure diagram in Figure 3-77:

- The types of day (RegularDayType) on which the service runs; for example, ‘Monday to Friday’, ‘Sunday’, or ‘Wednesday and Saturday’.
- The weeks of the month on which the service is operated for the given day types; for example, ‘first and third weeks of the month’. The PeriodicDayType further qualifies the RegularDayType, (for example, a market service that might run ‘Wednesdays and Saturdays, first and third weeks of the month’).
- The holiday or working day types of the serviced organisation for which the service runs (for example, ‘term times for City of London School for Girls’) – see ServicedOrganisation in Section 3.18.4 below. The ServicedOrganisationDayType further qualifies the periodic and regular day type. For example, ‘Wednesdays, during term times for City of London School for Girls’.

![UML Diagram of Normal Operation Profile](image)

3.18.3 Exceptional Operation – OperatingProfile

The OperatingProfile / Special group specifies the exceptional operating days of a service. It can be made up of two distinct elements, as shown in the UML structure diagram in Figure 3-78:
How the service operates on a bank holiday (BankHolidaysOperation). A number of different bank holiday day types are supported for both individual days and groups of days. For example ‘Does not run Christmas, New Year’s Day, Good Friday’, ‘Runs Bank Holiday Mondays’. Day types include moveable feasts, such as Easter Day, whose day may vary from year to year. The holidays on which the service does (inclusion) or does not (exclusion) run are specified separately.

Any special operating dates on which the service does (inclusion) or does not (exclusion) run (SpecialDaysOperation). Special days are always absolute calendar dates or calendar date ranges. For example ‘does not run 11/11/2005’. Special days override any Bank Holiday day types.

Special days are intended just to record impending exceptions and so should be used sparingly and not be used to cover long periods of non-operation. They should not be used with operating period dates far in the past or the future.

A statement that a service does not operate on specific days should not be interpreted as implying that it operates on all other days. Similarly a statement that a service runs on a particular day does not necessarily imply that it does not run on all other days.

Note that the exclusion and inclusion of special days of operation have different meanings (see also ‘General Principles for Using Operational Days’ below):

- The Special Operation profile days of non-operation i.e. exclusion should be interpreted as further constraining the days of week and month of the Normal Operating Profile. For example, if the Normal Profile specifies that a service runs ‘Monday to Friday’, and the Special Operation Profile specifies that the Service does not run on New Year’s Day, it will not run on New Year’s Day, whatever day of the week New Year’s Day occurs.

- The Special Operation profile days of explicit operation (i.e. inclusion) should be interpreted as being additive to the days of week and month of the Normal Operating Profile. For example, if
the Normal Operation Profile specifies that a service runs ‘Sunday’, and the Special Operation Profile specifies that the Service does run on New Year’s Day, then the service also runs to that timetable on New Year’s Day, regardless of the day of week New Year’s Day falls.

Thus a typical usage is to have a lower frequency Service timetable that is used for Sundays and Bank Holidays, and a regular timetable that is used for weekdays, except when the weekday is a Bank holiday.

3.18.4 Services that Run for Specific ServicedOrganisation Working Days

Operational day types can be specified in terms of the working days or holidays of specified organisations, for example schools. A hierarchical parent relationship can be used to specify that working days are derived from those of another organisation, for example a Local Education Authority (LEAs), with specific variations.

3.18.4.1 ServicedOrganisation Model Introduction

Figure 3-79 introduces the ServicedOrganisation:

- LEAs and their Schools are modelled in the schema by a ServicedOrganisation element. Each ServicedOrganisation may have a parent relationship (which should be acyclic) to another ServicedOrganisation.
- Patterns of WorkingDays and Holidays may be specified for ServicedOrganisations.
- Working days and holidays may be inherited from a parent organisation.
- Services and vehicle journeys may be associated with one or more organisations on the ServicedOrganisationDayType as part of the normal OperatingProfile.
Figure 3-79 – UML Diagram of Serviced Organisation: Intro
3.18.4.2 ServicedOrganisation Model Details

Figure 3-80 shows further details of Serviced Organisations.

**Figure 3-80 – UML Diagram of Serviced Organisation Days**
3.18.5 General Principles for Using Operational Days

The TransXChange model has capabilities to specify operational days at a number of different levels of discourse (Service, Journey Pattern, Vehicle Journey, Journey Interchange); and to state operational days in both relative and absolute terms; that is,

(i) As general day types, such as ‘Monday to Friday’, or ‘Christmas Day’ (using the OperatingProfile / Normal elements).

(ii) As absolute calendar dates, such as ‘5th - 7th August 2005’ (using the OperatingProfile / Special elements).

The different mechanisms can be combined to provide an overall set of operational conditions for a given vehicle journey that is to run on a given day of operation. When interpreting a schedule, a number of simplifying rules are followed for combining the various element types to avoid ambiguity. The following general principles are followed for the use of operational days:

1. **Elements specified for a given profile property at a lower level of discourse completely replace the equivalent element at a higher level.** For example, if a Journey level operating profile specifies days of operation as ‘Monday to Friday’ and a vehicle journey specifies ‘Saturday’, then the vehicle journey runs only on ‘Saturday’, not ‘Monday to Saturday’. Similar considerations apply to Serviced Organisation operating days for parent and child Service Organisation levels.

2. **Lower level of discourse overrides higher level for operational days.** In particular, any operational days specified for a specific vehicle journey take precedence over those specified over the journey pattern; and any for the journey pattern over those specified for the whole service. For example, a vehicle journey may state more restricted or more extensive operation days than the overall service. Table 3-17 shows the relative precedence of levels of discourse. Similarly Serviced Organisation properties override those of any parent organisation.

<table>
<thead>
<tr>
<th>Operational days Precedence (1 high)</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VehicleJourneyInterchange.</td>
</tr>
<tr>
<td>2</td>
<td>VehicleJourney.</td>
</tr>
<tr>
<td>3</td>
<td>JourneyPatternInterchange.</td>
</tr>
<tr>
<td>4</td>
<td>JourneyPattern.</td>
</tr>
<tr>
<td>5</td>
<td>Service.</td>
</tr>
</tbody>
</table>

Table 3-17 – Precedence of Entity Levels

3. **Exceptional operation overrides regular operation at any given level.** Thus OperatingProfile Special dates override any dates indicated by OperatingProfile Normal day types.

4. **Exclusion constrains, inclusion adds.** Special days of non-operating further restrict the normal profile; special days of operation are additional to the profile.

5. **Non-operation overrides operation.** If conflicting overlapping dates for operation and non-operation days are specified, the non-operation is assumed to be correct at any given level of discourse. This applies only within each level of discourse – operational days at a lower level override i.e. replace all non-operational days at a higher level.

6. **More specific day type overrides less specific day type.** At any given level of discourse, more specific normal OperatingProfile day type values qualify the less specific values – as shown in Table 3-18.

<table>
<thead>
<tr>
<th>Precedence (1 high)</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ServicedOrganisationDayType days of non-operation.</td>
</tr>
<tr>
<td>2</td>
<td>ServicedOrganisationDayType days of operation.</td>
</tr>
<tr>
<td>3</td>
<td>PeriodicDayType qualifier.</td>
</tr>
<tr>
<td>4</td>
<td>RegularDayType days.</td>
</tr>
</tbody>
</table>
Table 3-18 – Precedence of Normal Operation Day Types

3.18.6 Footnotes
The publisher generates explanatory footnotes from the day types, etc associated with each vehicle journey. These may arise from conditions specified at the Service, Journey Pattern or Vehicle Journey level. It may also append use defined notes.

3.18.6.1 Example - Footnotes
Figure 3-81 is an example of the automatically generated footnotes, output in order of generality.

Service operates from 22/09/2004 until further notice [P1]
Service runs Mondays on 3rd, 1st weeks in month [N1, N2]
Service runs during term times of Barham School [O1]
Service does not run during holidays of Barham School [O3]
Service does not run during working days of Bootle Works [O4]
Service runs during Christmas Day, Boxing Day, New Year’s Eve, Barday [20/12/2008] [H3, H4]
Service runs Bank Holiday Mondays, Good Friday, Fooday [12/12/2008] [H1, H2]
Service does not run 12/03/2008 until 12/04/2008 [S2]
Service runs 12/01/2008 until 12/02/2008 [S1]
Note: Electronic tickets accepted.

Figure 3-81 – Example of Footnotes

3.18.6.2 Footnote Variations
Table 3-19 shows the different footnotes generated for different TXC schema values (as updated for TXC v2.4).

<table>
<thead>
<tr>
<th>Period</th>
<th>Range</th>
<th>Argument</th>
<th>Note/ Footnotes</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Days</td>
<td>Regular Day Type</td>
<td>Start Date + End date</td>
<td>“Service operates from [99/99/99] to [99/99/99]”</td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td>DayOfWeek</td>
<td>Start Date + Open end date</td>
<td>“Service operates from[ 99/99/99] until further notice.”</td>
<td>P2</td>
</tr>
<tr>
<td>Normal Days</td>
<td>Periodic Day Type</td>
<td>WeekNumber</td>
<td>“Service runs Mondays, Thursdays on 3rd, 1st weeks in month”</td>
<td>N2</td>
</tr>
<tr>
<td>Holidays Only</td>
<td>HolidaysOnly</td>
<td></td>
<td>If specified at service level, amends heading to say: “Holidays Only”</td>
<td>N3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If at Vehicle Journey level adds footnote: “Only runs on specified Public Holiday”</td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>No values</td>
<td></td>
<td>If there is no profile generate the default values: “Service runs every day of the year including Christmas Day and Boxing Day.”</td>
<td>N0</td>
</tr>
<tr>
<td>Serviced Organisation</td>
<td>Days Of Operation</td>
<td>Holidays</td>
<td>“Service runs only during holidays of [Serviced Organisation Name]”</td>
<td>O1</td>
</tr>
<tr>
<td></td>
<td>Serviced- Organisation/Id</td>
<td></td>
<td>“Service runs only during working days of [Serviced Organisation Name].”</td>
<td>O2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If The Organisation is a school should be: “Service runs only during term time of [School Name]”.</td>
<td></td>
</tr>
<tr>
<td>Serviced Organisation</td>
<td>Days Of non- Operation</td>
<td>Holidays</td>
<td>“Service does not run during holidays of [Serviced Organisation Name].”</td>
<td>O3</td>
</tr>
<tr>
<td></td>
<td>Serviced- Organisation/Id</td>
<td></td>
<td>“Service does not run during working days of [Serviced Organisation Name].”</td>
<td>O4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the Organisation is a school should be “Service does not run during term time of [School Name].”</td>
<td></td>
</tr>
<tr>
<td>Exceptions</td>
<td>Special Days</td>
<td>Days Of Operation</td>
<td>“Service also runs [12/01/2008] until [12/02/2008].”</td>
<td>S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Days Of non- Operation</td>
<td>“Service does not [12/03/2008] until [12/04/2008].”</td>
<td>S2</td>
</tr>
</tbody>
</table>
### Holidays

<table>
<thead>
<tr>
<th>Days Of Operation</th>
<th>BankHolidays, Displacement-Holidays, EarlyRunOffDays</th>
<th>Service also runs [Christmas Day, Christmas Day Holiday, etc.] Unless BankHolidayOnly Service also runs [Christmas Day, Christmas Day Holiday, etc.]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OtherPublicHoliday</td>
<td>+Note</td>
<td></td>
</tr>
<tr>
<td>Days Of non-Operation</td>
<td>BankHolidays, Displacement-Holidays, EarlyRunOffDays +Note</td>
<td></td>
</tr>
<tr>
<td>OtherPublicHoliday</td>
<td>+Note</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Notes</th>
<th>Note</th>
<th>User defined text</th>
<th>User defined text</th>
</tr>
</thead>
</table>

**Table 3-19 – Footnotes generated for different inputs**

#### 3.18.6.3 Serviced Organisation Calendar Example

The following example shows calendars for two levels of service organisation - an LEA which sets default term times and a School which sets some specific overrides. The results are summarised in A Calendar of holidays can be associated with each service organisation. For example

```xml
<ServicedOrganisations>
  <!-- ======Parent LEA ========= -->
  <ServicedOrganisation>
    <OrganisationCode>SO_1</OrganisationCode>
    <Name>Bleakshire LEA</Name>
    <WorkingDays>
      <DateRange>
        <StartDate>2004-09-01</StartDate>
        <EndDate>2004-12-23</EndDate>
        <Description>Michaelmas Term</Description>
        <DateClassification>term</DateClassification>
      </DateRange>
      <DateRange>
        <StartDate>2005-01-01</StartDate>
        <EndDate>2005-04-30</EndDate>
        <Description>Easter Term</Description>
        <DateClassification>term</DateClassification>
      </DateRange>
      <DateRange>
        <StartDate>2005-04-02</StartDate>
        <EndDate>2005-07-23</EndDate>
        <Description>Summer Term</Description>
        <DateClassification>term</DateClassification>
      </DateRange>
    </WorkingDays>
    <Holidays>
      <DateRange>
        <StartDate>2004-11-11</StartDate>
        <EndDate>2004-11-11</EndDate>
        <Description>Inset day</Description>
        <DateClassification>inset</DateClassification>
      </DateRange>
      <DateRange>
        <StartDate>2004-11-01</StartDate>
        <EndDate>2004-11-07</EndDate>
        <Description>Autumn Half term</Description>
        <DateClassification>holiday</DateClassification>
      </DateRange>
      <DateRange>
        <StartDate>2005-02-07</StartDate>
        <EndDate>2005-02-14</EndDate>
        <Description>Spring Half term</Description>
        <DateClassification>holiday</DateClassification>
      </DateRange>
    </Holidays>
  </ServicedOrganisation>
</ServicedOrganisations>
```
Figure 3-82 – Example LEA Calendar

```xml
<!DOCTYPE Dotheboy Organisation -->
<ServicedOrganisation>
  <OrganisationCode>SO_2</OrganisationCode>
  <Name>Dothboy's Academy</Name>
  <WorkingDays>
    <DateRange>
      <StartDate>2005-04-01</StartDate>
      <EndDate>2005-07-30</EndDate>
      <Description>Summer Term</Description>
    </DateRange>
    <Holidays>
      <DateRange>
        <StartDate>2004-11-12</StartDate>
        <EndDate>2004-11-12</EndDate>
        <Description>Inset day</Description>
        <DateClassification>inset</DateClassification>
      </DateRange>
      <Holidays>
        <DateRange>
          <StartDate>2004-11-11</StartDate>
          <EndDate>2004-11-11</EndDate>
          <Description>Inset day</Description>
          <DateClassification>inset</DateClassification>
        </DateRange>
      </Holidays>
    </Holidays>
  </WorkingDays>
</ServicedOrganisation>
</ServicedOrganisations>
```

Figure 3-83 – Example School Calendar

Table 3-20 & Table 3-21 show possible output

<table>
<thead>
<tr>
<th>Serviced Organisation:</th>
<th>SO_1</th>
<th>SO_2</th>
<th>Bleakshire LEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>To</td>
<td>From</td>
<td>To</td>
</tr>
<tr>
<td>Working Days</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-09-01</td>
<td>2004-12-30</td>
<td>Michaelmas Term</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-01-01</td>
<td>2005-04-30</td>
<td>Easter Term</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-04-01</td>
<td>2005-07-29</td>
<td>Summer Term</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-04-25</td>
<td>2005-06-01</td>
<td>Summer Hal term</td>
</tr>
<tr>
<td>Holidays</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-11-11</td>
<td>2004-11-11</td>
<td>Inset day</td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-11-01</td>
<td>2004-11-07</td>
<td>Autumn Hal term</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-02-07</td>
<td>2005-02-14</td>
<td>Spring Hal term</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-05-25</td>
<td>2005-06-01</td>
<td>Summer Hal term</td>
</tr>
</tbody>
</table>

Table 3-20 – Example Serviced Organisation Particulars for LEA

<table>
<thead>
<tr>
<th>Serviced Organisation</th>
<th>SO_2</th>
<th>SO_1</th>
<th>Dothboy's Academy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent</td>
<td></td>
<td>SO_1</td>
<td>Bleakshire LEA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Working Days:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-09-01</td>
<td>2004-12-23</td>
</tr>
<tr>
<td>SO_1</td>
<td>2005-01-01</td>
<td>2005-04-30</td>
</tr>
<tr>
<td>SO_2</td>
<td>2005-04-01</td>
<td>2005-07-30</td>
</tr>
<tr>
<td>Holidays</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-11-12</td>
<td>2004-11-12</td>
</tr>
<tr>
<td>SO_1</td>
<td>2004-11-01</td>
<td>2004-11-07</td>
</tr>
</tbody>
</table>
### Table 3-21 – Example Serviced Organisation Particulars for School

<table>
<thead>
<tr>
<th>SO_1</th>
<th>2005-02-07</th>
<th>2005-02-14</th>
<th>Spring Half term</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO_1</td>
<td>2005-05-25</td>
<td>2005-06-01</td>
<td>summer Half term</td>
</tr>
</tbody>
</table>
3.18.7 Frequently Services

A *Frequent Service* is a service that meets the regulatory requirements for being classified as a frequent service, i.e. that runs to a frequency of every ten minutes or less in accordance with the Statutory Requirement, and that is to be formally registered as constituting a *Frequent Service*, as indicated by a *FrequentService* flag. Note that in addition, a minimum and maximum time gap between services operating as a frequency service can also be specified using the *MinimumFrequency* and *MaximumFrequency* elements.

Journeys which comprise a frequent service do not have to run at an absolutely regular frequency interval - they could be quite variable, such as every 2 - 7 minutes, as long as no service interval exceeds 10 minutes between consecutive journeys. The service should be described as running to its lowest available frequency e.g. ‘Frequent service at least every 7 mins.’

3.18.8 Frequency Based Services

Independently of whether the service is legally a *Frequent Service*, the TransXChange schema supports a *Frequency Based Service* definition; that is, a service that runs to a regular frequency, for example ‘every 5 minutes’ or ‘every 15 minutes’, rather than to a specific timetable (and which may or may not be a statutory *Frequent Service* – in which case it would be phrased ‘Frequent service at least every 5 mins.’).

The frequency pattern of a *VehicleJourney* is described by a *Frequency* element which holds elements giving the frequency of the service, and an end time. Frequencies may be specified either as an *Interval* of minutes (see Table 3-23), or as a collection of *MinutesPastTheHour* instances (see Table 3-24).

The *TransXChange* schema allows the departure times for vehicle journeys occurring at regular intervals to be coded efficiently as a single vehicle journey, with a frequency to be repeated and an end-time (i.e. the last departure time that follows the standard pattern). Such journeys may or may not be part of a *Frequent Service*. Using the mechanism, just one vehicle journey is needed in the document rather than, say, many journeys that are identical but for the departure time. The *TransXChange* Publisher will then generate the necessary

3.18.8.1 Frequency Described by Interval

Table 3-22 shows a frequency based timetable described using a single journey and Frequency interval. Only the initial journey of a period of frequency based service need be explicitly given, so the entire timetable can be described by a single vehicle journey, as per column #1, together with a *ScheduledFrequency* (15 minutes) and an *EndTime* (12:04).

<table>
<thead>
<tr>
<th>DepartureTime</th>
<th>9:02</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScheduledFrequency</td>
<td>15</td>
</tr>
<tr>
<td>EndTime</td>
<td>12:02</td>
</tr>
<tr>
<td>Grub Street</td>
<td>9:02</td>
</tr>
<tr>
<td>Tin Pan Alley</td>
<td>9:12</td>
</tr>
<tr>
<td>Sinister Street</td>
<td>9:32</td>
</tr>
</tbody>
</table>

Table 3-22 – Frequency Service Timetable representation: Interval

Table 3-23 shows this as published - the Publisher generates the additional columns.

| J1 | J1 | #1 | #2 |
| J1 | 9:02 | Then every 15 minutes until 12:02 |
| Grub Street | 9:02 | 15 minutes |
| Tin Pan Alley | 9:12 |
| Sinister Street | 9:32 |

Table 3-23 – Example Frequent Service Timetable presentation: Interval
### 3.18.8.2 Departure Described by Minutes Past Hour

*Table 3-24* shows an example of a service described using minutes past the hour. This can be used to describe services that don’t run at regular intervals columns #1, #2 and #3 are all described by a single vehicle journey with a start and end time, and a frequency stated as two different minutes past the hour. Column #4 is a new journey.

<table>
<thead>
<tr>
<th>Start time</th>
<th>Minutes (Past Hour)</th>
<th>End Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:02</td>
<td>12 30</td>
<td>12:02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grub Street</td>
<td>9:02</td>
<td>12</td>
<td>30</td>
</tr>
<tr>
<td>Tin Pan Alley</td>
<td>9:12</td>
<td>22</td>
<td>40</td>
</tr>
<tr>
<td>Sinister Street</td>
<td>9:32</td>
<td>42</td>
<td>00</td>
</tr>
</tbody>
</table>

**Table 3-24 – Example Frequent Service Timetable presentation: Minutes**

### 3.18.8.3 Frequency Described on Multiple Individual Journeys – Merged Frequency

For some purposes it is useful to supply information about every single journey making up a *Frequent service*, for example, so as to be able to specify operational Block and Run information on journeys for AVL systems. Within the period of frequent operation these multiple, individually timed journeys can still be published as a single Frequency Group, that is a column of start times and a column giving the frequency, rather than separate columns for each journey. The TransXChange publisher will perform this merging of separate journeys as follows:

- If successive vehicle journeys are (a) flagged as *Frequency Based* and (b) have the same *EndTime* as the previous journey, then they will be collapsed into a single Frequency column.
- The indicated frequency values should normally also be the same for all journeys (*Scheduled Frequency, Minimum Frequency, Maximum Frequency*). If they differ the values from the first journey will be used and a diagnostic will be added to the validation report.
- If any frequent services are provided as individual journeys for a frequent service in a document then all the individual journeys should be provided.
- Note that the individual vehicle journeys themselves do not have to be at exactly regular intervals.
- The merging of journeys by the publisher can be suppressed using the *mergeFrequencyJourneys* option. (This is useful for those that wish to see the underlying data).

Thus one might have many journeys, the journey intervals are all slightly different as indicated by different start times, but less than the 10 minutes but rather than construing them as separate journeys in Table 3-25 below.

<table>
<thead>
<tr>
<th>Departure time</th>
<th>9:02</th>
<th>9:10</th>
<th>9:16</th>
<th>9:21</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScheduledFrequency</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>MinimumFrequency</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>MaximumFrequency</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>EndTime</td>
<td>12:02</td>
<td>12:02</td>
<td>12:02</td>
<td>12:02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grub Street</th>
<th>9:02</th>
<th>9:10</th>
<th>9:16</th>
<th>9:21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tin Pan Alley</td>
<td>9:12</td>
<td>9:20</td>
<td>9:26</td>
<td>9:31</td>
</tr>
<tr>
<td>Sinister Street</td>
<td>9:32</td>
<td>9:40</td>
<td>9:36</td>
<td>9:51</td>
</tr>
</tbody>
</table>

**Table 3-25 – Multi-journey Representation of Frequency Based journeys**

The Publisher would present the journeys more concisely as in Table 3-26. (The actual text in column #2 will vary as per *Table 3-31*. )

| j1 | j2 to j(n-1) | j(n) |
#1 | #2 | #3
---|---|---
Grub Street | 9:02 | Frequent service at an interval of no more than 7 mins | 12:02
Tin Pan Alley | 9:12 | 12:12 |
Sinister Street | 9:32 | 12:32 |

Table 3-26 – Merged presentation of separate Frequency journeys having the same frequencies and end time

### 3.18.8.4 Multi-journey to single group, multiple frequencies

If the frequency changes between journeys, as in Table 3-27.

<table>
<thead>
<tr>
<th>Departure time</th>
<th>ScheduledFrequency</th>
<th>EndTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:02</td>
<td>15</td>
<td>11:02</td>
</tr>
<tr>
<td>9:17</td>
<td>15</td>
<td>11:20</td>
</tr>
<tr>
<td>9:32</td>
<td>15</td>
<td>11:20</td>
</tr>
<tr>
<td>9:47</td>
<td>15</td>
<td>16:00</td>
</tr>
</tbody>
</table>

Table 3-27 – Multi-journey Representation of Two Frequencies

The Publisher can add additional columns to describe the change in frequency as in Table 3-28. The additional column would be triggered by separate EndTime values (11:02, then 16:00), not by the separate ScheduledFrequency value. If the end time is the same, only a single column will be shown with the first scheduled frequency.

<table>
<thead>
<tr>
<th>J1</th>
<th>J2 to J(n-1)</th>
<th>J(n)</th>
<th>J(m)</th>
<th>J(m+1) to J(p-1)</th>
<th>J(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grub Street</td>
<td>9:02</td>
<td>Then every 15 minutes</td>
<td>11:02</td>
<td>11:20</td>
<td>16:00</td>
</tr>
<tr>
<td>Tin Pan Alley</td>
<td>9:12</td>
<td>About every 5 minutes</td>
<td>About every 5 minutes</td>
<td>About every 8 minutes</td>
<td></td>
</tr>
<tr>
<td>Sinister Street</td>
<td>9:32</td>
<td>About every 6 minutes</td>
<td>About every 6 minutes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3-28 – Merged presentation of separate Journeys with different frequencies

### 3.18.9 Partial Frequency Based Services (+TXC v2.4)

Some services are presented as being timetabled to absolute times at certain stops and as Frequency Based for other parts of the journey. This may be specified by an override Frequency Interval on the individual VehicleJourneyStopUsage of the VehicleJourneyTimingLink of just those stops that are to be shown as frequency based. This feature can only be used if journeys are individually specified (See Merged Frequency Journeys above).

<table>
<thead>
<tr>
<th>Departure time</th>
<th>PartialFrequency</th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:02</td>
<td>false</td>
<td>9:10</td>
<td>9:16</td>
<td>9:16</td>
</tr>
<tr>
<td>9:10</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3-29 shows the representation of a three Partial Frequency Journeys:

<table>
<thead>
<tr>
<th>Departure time</th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:02</td>
<td>9:10</td>
<td>9:16</td>
<td>9:21</td>
</tr>
<tr>
<td>9:10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3-29 – Representation of Partial Frequency Based journeys

Table 3-30 shows the resulting presentation of the Partial Frequency Journeys in a matrix:

<table>
<thead>
<tr>
<th>Departure time</th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
<th>J4</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:02</td>
<td>9:10</td>
<td>9:16</td>
<td>9:21</td>
<td></td>
</tr>
<tr>
<td>9:10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9:16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3-30 – Presentation of Partial Frequency Based journeys
3.18.9.1 Text Descriptions for Frequency service

The text caption used in a column to describe the frequency is generated from the values of the ScheduledFrequency MinimumFrequency, MaximumFrequency associated with the journey, as per Table 3-31. For registrations the less informative statutory definition is used.

<table>
<thead>
<tr>
<th>Case</th>
<th>Frequent Service</th>
<th>Scheduled-Frequency Interval (mins)</th>
<th>Minimum-Frequency Interval (mins)</th>
<th>Maximum Frequency Interval (mins)</th>
<th>Result Phrase to show in matrix column for NON-REGISTRATION details</th>
<th>Result Phrase to show in matrix column for REGISTRATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>true</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>then about every ([x]) minutes until</td>
<td>Frequent service at least every 10 mins until</td>
</tr>
<tr>
<td>2</td>
<td>true</td>
<td>x</td>
<td>m</td>
<td>-</td>
<td>then at ([m-x]) minutes intervals until</td>
<td>Frequent service at least every 10 mins until</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>x</td>
<td>-</td>
<td>n</td>
<td>then at intervals of no more than ([n]) mins until</td>
<td>Frequent service at least every 10 mins until</td>
</tr>
<tr>
<td>4</td>
<td>true</td>
<td>x</td>
<td>m</td>
<td>n</td>
<td>then at ([m-n]) minutes intervals until</td>
<td>Frequent service at least every 10 mins until</td>
</tr>
<tr>
<td>5</td>
<td>false</td>
<td>x</td>
<td>-</td>
<td>-</td>
<td>then about every ([x]) minutes until</td>
<td>Then every ([x]) mins until</td>
</tr>
<tr>
<td>6</td>
<td>false</td>
<td>x</td>
<td>m</td>
<td>-</td>
<td>then about every ([m-x]) minutes until</td>
<td>(Then every ([x]) mins until)</td>
</tr>
<tr>
<td>7</td>
<td>false</td>
<td>x</td>
<td>-</td>
<td>n</td>
<td>then at intervals of no more than ([n]) mins until</td>
<td>(Then every ([x]) mins until)</td>
</tr>
<tr>
<td>8</td>
<td>false</td>
<td>x</td>
<td>m</td>
<td>n</td>
<td>then at ([m-n]) minutes intervals until</td>
<td>(Then every ([x]) mins until)</td>
</tr>
</tbody>
</table>

Table 3-31 – Frequency service Text Descriptions

3.18.10 Service Operational Days

Using the principles given above, we can summarise the use of the operational day elements shown in Figure 3-77 to specify the operational days of a Service as follows:

1. Each Service has an OperatingPeriod defining its overall start and end dates. All operational dates must fall within this period. The end date may be open.
   a. Operating period: The start date should be the first date of service operation or a date very close to it, if the start date is being aligned with other service changes. For example, it is in order for a Tuesday only service to have an operation period start on the preceding Sunday. A dummy date value e.g. 0000-00-00 or 1901-00-00 should not be used.
   b. StartDate: With the possible exception of certain London and Northern Ireland lines, no start date for a bus service in the UK should have a start date earlier than 26 October 1986.
   c. EndDate: The end date should be omitted if no end date is known, since unrealistic future dates such as year 9999, can create a misleading impression of certainty
   d.

2. Each Service can have a default OperatingProfile describing its operation: the Service / OperatingProfile normal elements are used to provide default values for all vehicle journeys of the service. Regular days can be specified in any or all of three ways (which can be combined together):
   a. RegularDayType: Any combination of days of the week on which the service runs. Defaults to MondayToSunday if not specified.
   b. PeriodicDayType: Additional qualifier of specific weeks of month on which the regular service runs.
   c. ServicedOrganisationDayType: Dates defined by the working days or holidays of a named organisation, such as a school or Local Education Authority. ServicedOrganisations can also be used to represent other types of organisation such as Works, Football Stadia.

Special days can be specified in two ways (which can be combined together):
d. **BankHolidayOperation:** Specific named *BankHoliday* day types (for example, *ChristmasDay*; see *BankHoliday* element, or instances of one-off holidays (such as, say, a Silver Jubilee) described by an *OtherPublicHoliday* instance, are assigned to one of two categories:
   i. *Days of operation.* Bank holiday days for which the service operates. If the specified days of operation overlap with days of non-operation, the days of non-operation take precedence.
   ii. *Days of non-operation.* Bank holiday days for which the service does not operate. If the specified days of non-operation overlap with days of operation, the days of non-operation take precedence over days of operation.

e. **SpecialDaysOfOperation:** Specific *DateRange* elements, assigned to one of two categories:
   i. *Days of operation.* If the specified days of operation overlap with days of non-operation, the days of non-operation take precedence.
   ii. *Days of non-operation.* If the specified days of non-operation overlap with days of operation, the days of non-operation take precedence over days of operation.

3. Each *JourneyPattern* can also have a specific *OperatingProfile*, describing its individual operational days. The profile is made up of the same elements as the Service profile. Any values override the service level values.

4. Each *VehicleJourney* can have a specific *OperatingProfile*, describing its individual operational days. The profile is made up of the same elements as the Service profile. Any values override any service and journey pattern level values.

5. Interchange *ValidityPeriod:* As a further complication, a *ValidityPeriod* may be specified for individual interchanges at both the *JourneyPatternInterchange* and the *VehicleJourneyInterchange* level.
   a. Use of the Interchange is only valid during the specified validity period. The connection will not be available to the inbound vehicle journey except during the validity period.
   b. Any *ValidityPeriod* specified at the *VehicleJourneyInterchange* level overrides any *ValidityPeriod* specified at the *JourneyPatternInterchange* level.

*Table 14-5* in Section 14.3; *‘Precedence Rules for Dates’*, summarises the conditions for specifying dates.

3.18.11 Structure Example of Schedule with Operational Day Exceptions

For an Example of a service using complex dates and times see the *Interchange Example*.

3.19 Calendar Model

In order to relate Operating days to DayTypes a *Calendar* can be used. A Calendar (provides a list of days that can be assigned to specific named day types.

3.19.1.1 Calendar Model Introduction

Figure 3-84 introduces the *Calendar* Model. A *Calendar* is made up of one or more *OperatingDays*. Each has a date and may be assigned to a day type (e.g. *MondayToFriday, Sunday*, etc)
3.19.1.2 Calendar Model Details

Figure 3-85 shows further details of the Calendar Model.

3.19.2 Calendar Model Hierarchy

Figure 3-86 shows the inheritance hierarchy for the Calendar model elements.
**Figure 3-86 – UML Diagram of Calendar Element Hierarchy**

**ServiceCalendar**
- Name : MultilingualString [0..1]
- From : date
- To : date
- Id : CalendarIdType

**ServiceDayAssignment**
- NoService : EmptyType [0..1]
- UnspecifiedService : EmptyType [0..1]
- VehicleJourneyRefs : VehicleJourneyCodeType [0..*]
3.20 Summary of TransXChange Entities and Identifiers

Table 3.2 summarises the significant entities of the TransXChange model. It also shows the identifiers used for each element and their scope (which in all cases must be unique within a document). The element identifiers fall into three scope groups:

- **External Codes forming part of well-defined national data systems** (‘A’). For example the **AtcoCode**, as defined in the NaPTAN data set. External codes are modelled as elements. These identifiers will always remain the same on repeated reissues of a given schedule as a TransXChange document.

- **External Codes forming part of arbitrary data systems**. (‘B’). External codes are modelled as XML elements, and their names generally end in either ‘Code’ or ‘Number’. These identifiers will normally remain the same on repeated reissues of a given schedule as a TransXChange document.

- **Internal Identifiers** used to identify objects locally within a document (‘C’). Internal identifiers are modelled as an id attribute on the entity element. Most id attributes are of type IDType. It is up to the application to decide whether internal identifiers should persist between different versions of a document. Typically there is no guarantee that these will remain the same on repeated reissues of a given schedule as a TransXChange document, though implementers are free to make them so if they wish.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Identifier</th>
<th>Type</th>
<th>Required</th>
<th>Name</th>
<th>Has Private Code</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopPoint</td>
<td>AtcoCode,</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>Yes</td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>NaptanCode,</td>
<td>Element</td>
<td>O</td>
<td></td>
<td>Yes</td>
<td>A-National</td>
</tr>
<tr>
<td>StopArea (Cluster)</td>
<td>StopAreaCode,</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>Yes</td>
<td>A-National</td>
</tr>
<tr>
<td>AdministrativeArea</td>
<td>AdministrativeAreaCode</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>A-National</td>
</tr>
<tr>
<td>NptgLocality</td>
<td>NptgLocalityCode</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>Yes</td>
<td>A-National</td>
</tr>
<tr>
<td>ServiceOrganisation</td>
<td>OrganisationCode</td>
<td>Element</td>
<td>R</td>
<td>OrganisationCode</td>
<td>TXC2.4</td>
<td>B-Various</td>
</tr>
<tr>
<td>Garage</td>
<td>GarageCode,</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>B-Operator</td>
</tr>
<tr>
<td>Service</td>
<td>ServiceCode,</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>Yes</td>
<td>B-Operator</td>
</tr>
<tr>
<td></td>
<td>TicketMachineServiceCode</td>
<td>Element</td>
<td>E</td>
<td>TicketMachineServiceCode</td>
<td>Yes</td>
<td>C-Document</td>
</tr>
<tr>
<td>Operator</td>
<td>id</td>
<td>Element</td>
<td>O</td>
<td>NationalOperatorCode</td>
<td>Yes (TXC2.4)</td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>OperatorCode</td>
<td>Element</td>
<td>O</td>
<td></td>
<td></td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>LicenceNumber</td>
<td>Element</td>
<td>O</td>
<td></td>
<td></td>
<td>A-National</td>
</tr>
<tr>
<td>Registration</td>
<td>VosaRegistrationNumber</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>/ TanCode</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>/ LicenceNumber</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>A-National</td>
</tr>
<tr>
<td></td>
<td>/ RegistrationNumber</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>A-National</td>
</tr>
<tr>
<td>Line</td>
<td>LineName</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>No</td>
<td>B-Service</td>
</tr>
<tr>
<td></td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td>LineName</td>
<td></td>
<td>C-Document</td>
</tr>
<tr>
<td>Route</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>Yes</td>
<td>C-Document</td>
</tr>
<tr>
<td>RouteSection</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>RouteLink</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td>RouteLink</td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>JourneyPattern</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>Yes</td>
<td>C-Document</td>
</tr>
<tr>
<td>JourneyPatternSection</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>JourneyPatternTimingLink</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>JourneyPatternStopUsage</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>JourneyPatternInterchange</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>VehicleJourney</td>
<td>VehicleJourneyCode</td>
<td>Element</td>
<td>R</td>
<td></td>
<td>Yes</td>
<td>B-Service</td>
</tr>
<tr>
<td>VehicleJourneyTimingLink</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>VehicleJourneyStopUsage</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>VehicleJourneyInterchange</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>DeadRun</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>PositioningLink</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>LayoverPoint</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>No</td>
<td>C-Document</td>
</tr>
<tr>
<td>AccessVehicleEquipment</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td>AccessVehicleEquipment</td>
<td>TXC2.5</td>
<td>C-Document</td>
</tr>
<tr>
<td>AssistanceService</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td>AssistanceService</td>
<td>TXC2.5</td>
<td>C-Document</td>
</tr>
<tr>
<td>ServiceFacilitySet</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>TXC2.5</td>
<td>C-Document</td>
</tr>
<tr>
<td>Calendar</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>TXC2.4</td>
<td>C-Document</td>
</tr>
<tr>
<td>Contributor</td>
<td>id</td>
<td>Attribute</td>
<td>O</td>
<td></td>
<td>TXC2.4</td>
<td>C-Document</td>
</tr>
</tbody>
</table>
The uniqueness scope of many identifiers is formally defined in the TransXChange schema by XML keyref constraints. See ‘Integrity Rules’ in Section 14.

### 3.20.1 Summary of TransXChange Entity Identifier Types

Table 3-33 shows the data types for the main identifiers used in the TransXChange model.

<table>
<thead>
<tr>
<th>Name</th>
<th>Base Type (2.4)</th>
<th>Former Type (TXC 2.2)</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>AtcoCodeType</td>
<td>Xsd:normalizedString</td>
<td>Xsd:string</td>
<td></td>
</tr>
<tr>
<td>NaptanCodeType</td>
<td>Xsd:normalizedString</td>
<td>Xsd:string</td>
<td>Yes: 1 or alpha8 p[prefix]</td>
</tr>
<tr>
<td>AdministrativeAreaCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td>Restricted Code set</td>
</tr>
<tr>
<td>AtcoAreaCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td>Restricted Prefix</td>
</tr>
<tr>
<td>NptgLocalityCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>OrganisationCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>GarbageCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>ServiceCodeType</td>
<td>Xsd:normalizedString</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>TicketMachineServiceCodeType</td>
<td>Xsd:normalizedString</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>TicketMachineJourneyCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>LicenceNumber</td>
<td>Xsd:normalizedString</td>
<td>Xsd:string</td>
<td></td>
</tr>
<tr>
<td>VosaRegistrationNumber</td>
<td>na</td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>RegionCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>RegionShortCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>NptgDistrictCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>NaptanPrefixType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>PlusbusZoneCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>CallCentreCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
</tbody>
</table>

3.20.2 Summary of NPTG Entity Identifier Types

Table 3-34 shows the data types for the NPTG identifiers used in the TransXChange model though the incorporation of NaPTAN Stop Points and Stop Areas (+TXC 2.4).

<table>
<thead>
<tr>
<th>Name</th>
<th>Base Type (2.4)</th>
<th>Type (2.2)</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegionCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>RegionShortCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>NptgDistrictCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td>Restricted Code set</td>
</tr>
<tr>
<td>NptanPrefixType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td>Restricted Prefix</td>
</tr>
<tr>
<td>PlusbusZoneCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
<tr>
<td>CallCentreCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
<td></td>
</tr>
</tbody>
</table>

3.20.3 Private codes

For a number of semantically significant elements, an additional PrivateCode element is supported. The PrivateCode facilitates the general purpose exchange of data in TransXChange format, as it
allows instances to be annotated with the alternative identifier, so as to allow the unambiguous reconciliation of element identity between different computer systems on round trip exchanges. Table 3-32 also indicates the elements that can have a **PrivateCode**.

Note: Private codes are used in preference to XML ANY element types, as the latter cause a reduction in the efficacy of some commonly used validators.

### 3.20.4 Referencing Elements

A systematic convention is used to show the implementation of relationships (other than inline containment) between elements. For each entity that is referenced, a **RefStructure** is defined (based on the same type as the identifier of the referenced element), and this structure is used to type all references. This helps when reading the schema — if you see an element with REF on it, you know it implements a relationship with another entity. Table 3-35 lists the elements that are referenced in various relationships.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Identifier</th>
<th>Type</th>
<th>Scope</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopPoint</td>
<td>StopPointRef</td>
<td>AtcoCodeType</td>
<td>NaPT</td>
<td>2.1</td>
</tr>
<tr>
<td>StopArea</td>
<td>StopAreaRef</td>
<td>StopAreaCodeType</td>
<td>NaPT</td>
<td>2.1</td>
</tr>
<tr>
<td>AdministrativeArea</td>
<td>AdministrativeAreaRef</td>
<td>AdminAreaCodeType</td>
<td>NPTG</td>
<td>2.1</td>
</tr>
<tr>
<td>NptgLocality</td>
<td>NptgLocalityRef</td>
<td>NptgLocalityCodeType</td>
<td>NPTG</td>
<td>2.1</td>
</tr>
<tr>
<td>ServicedOrganisation</td>
<td>ServicedOrganisationRef</td>
<td>OrganisationCodeType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Contributor</td>
<td>OrganisationRef</td>
<td>OrganisationCodeType</td>
<td>TXC</td>
<td>2.4</td>
</tr>
<tr>
<td>Service</td>
<td>ServiceRef</td>
<td>ServiceCodeType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Operator</td>
<td>OperatorRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Registration</td>
<td>RegistrationRef</td>
<td>VosaRegistrationNumberStructure</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Line</td>
<td>LineRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Route</td>
<td>RouteRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>RouteSection</td>
<td>RouteSectionRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>RouteLink</td>
<td>RouteLinkRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPattern</td>
<td>JourneyPatternRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternSection</td>
<td>JourneyPatternSectionRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternTimingLink</td>
<td>JourneyPatternTimingLinkRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternStopUsage</td>
<td>JourneyPatternStopUsageRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternInterchange</td>
<td>JourneyPatternInterchangeRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>VehicleJourney</td>
<td>VehicleJourneyRef</td>
<td>VehicleJourneyCode</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>VehicleJourneyTimingLink</td>
<td>VehicleJourneyTimingLinkRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>VehicleJourneyStopUsage</td>
<td>VehicleJourneyStopUsageRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>VehicleJourneyInterchange</td>
<td>VehicleJourneyInterchangeRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>DeadRun</td>
<td>DeadRunRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>PositioningLink</td>
<td>PositioningLinkRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>LayoverPoint</td>
<td>LayoverPointRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Location</td>
<td>LocationRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Calendar</td>
<td>CalendarRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.4</td>
</tr>
<tr>
<td>DataRight</td>
<td>DataRightRef</td>
<td>idType</td>
<td>TXC</td>
<td>2.4</td>
</tr>
<tr>
<td>SupportingDocument</td>
<td>--</td>
<td>DocumentUri</td>
<td>TXC</td>
<td>2.1</td>
</tr>
<tr>
<td>Note</td>
<td>--</td>
<td>NoteCode</td>
<td>TXC</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Table 3-35 – References to Entities in the TransXChange Model

### 3.20.5 Summary of TransXChange Ancillary Identifiers

Table 3-32 summarises the ancillary external entities referenced by the **TransXChange** model. It also shows the identifiers used for each element. The scope is local to the document in all cases.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Identifier</th>
<th>Base Type 2.4</th>
<th>Base Type 2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopPlate</td>
<td>PlateCodeType</td>
<td>Xsd:normalizedString</td>
<td>Xsd:normalizedString</td>
</tr>
<tr>
<td>StopCleardown</td>
<td>CleardownCodeType</td>
<td>Xsd:positiveInteger</td>
<td>Xsd:positiveInteger</td>
</tr>
<tr>
<td>DutyCrew</td>
<td>DutyCrewCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
</tr>
<tr>
<td>TicketMachineJourney</td>
<td>TicketMachineJourneyCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
</tr>
<tr>
<td>TicketMachineService</td>
<td>TicketMachineServiceCodeType</td>
<td>Xsd:NMTOKEN</td>
<td>Xsd:NMTOKEN</td>
</tr>
<tr>
<td>FareStage</td>
<td>FareStageNumberType</td>
<td>Xsd:nonNegativeInteger</td>
<td>Xsd:nonNegativeInteger</td>
</tr>
</tbody>
</table>
3.21 Common Data types

3.21.1 XML Base Types

Figure 3-87 summarises the base XML data types used in NaPTAN & TransXChange.

3.21.2 General Utility Types

Figure 3-88 summarises the common utility XML data types used in NaPTAN & TransXChange.
3.22 Data Rights Model

A TransXChange document can specify the allowed use of the data.

3.22.1 Data Rights Model - Introduction

Figure 3-89 introduces the rights model: a DataRight can be associated with every VersionedObject. Data Rights may be held by an Operator, or by a Contributor Organisation.

Figure 3-89 – UML Diagram of Rights Model: Introduction
3.22.2 Data Rights Model - Introduction

Figure 3-89 shows further details of the DataRight model: It is possible to specify Copyright, Terms of Use and a Data Policy – including whether the data is subject to Freedom of Information act, with each element. The DataRight of a parent element is deemed to apply to its children.
4 WORKED EXAMPLE OF A TRANSXCHANGE SCHEDULE

This section provides a basic introductory example of using the main TransXChange elements. For more complex examples, refer to Section 5.

4.1 Worked Example: Bus Timetable

The elements of a TransXChange StandardService are illustrated using the fictional timetable shown in Table 4-1:

<table>
<thead>
<tr>
<th>Service:</th>
<th>1, Suborn - Beall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin:</td>
<td>Suborn, Bus Station</td>
</tr>
<tr>
<td>Destination:</td>
<td>Beall, Bus Exchange</td>
</tr>
</tbody>
</table>

Notes: Valid from 5 February until further notice

<table>
<thead>
<tr>
<th>Mondays to Fridays / Service on Mayday</th>
<th>Operator:</th>
<th>Journey:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ACO</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>RED</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>ACO</td>
<td>43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Garden Village, Shop</td>
<td></td>
<td>16:46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robridge, Plough</td>
<td></td>
<td>16:26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Barford, Red Lion</td>
<td>Depart:</td>
<td>16:09</td>
<td>16:29</td>
<td>16:49</td>
<td>16:54</td>
<td>17:09</td>
<td>17:29</td>
<td>17:49</td>
</tr>
<tr>
<td>Egham, Golden Lion</td>
<td></td>
<td>16:12</td>
<td>16:52</td>
<td>16:57</td>
<td>17:12</td>
<td>17:52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Godhill Church</td>
<td></td>
<td>16:15</td>
<td>16:55</td>
<td>17:00</td>
<td>17:15</td>
<td>17:55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beall, Exchange</td>
<td></td>
<td>16:32</td>
<td>16:52</td>
<td>17:12</td>
<td>17:32</td>
<td>17:52</td>
<td>18:12</td>
<td></td>
</tr>
<tr>
<td>Beall Business Park, Shell</td>
<td></td>
<td>16:53</td>
<td>17:16</td>
<td>17:53</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beall, Bus Station</td>
<td>Arrive:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17:17</td>
</tr>
</tbody>
</table>

Table 4-1 – Worked Example: Bus Timetable

4.2 Worked Example: Service Components

To encode the example, we use a StandardService comprising:

- A LicensedOperator, who registers the service.
- A Registration, recording the statutory registration of the service with a TAN.
- A StandardService, recording the schedule of a fixed route service.
- A Route, over which the service runs from StopPoint to StopPoint.
- A JourneyPattern, describing a general journey as a sequence of timing links.
- A collection of VehicleJourney instances, describing individual journeys as timing link sequences, and the departure times at which they run. Each VehicleJourney is based on a JourneyPattern.

4.3 Worked Example: Operator

Two types of operator can be defined for a service: the RegisteredOperator who registers the service, and one or more AssociatedOperator instances, who may perform subsidiary roles.

In the example case the registering LicensedOperator is ‘ACO’, and the single associated Operator is ‘RED’, who runs one particular journey on behalf of ‘ACO’.
4.4 Worked Example: Registration

The Registration holds administrative details of the service registration, such as SubmissionDate, SubmissionAuthor, and the TrafficAreas with full or partial responsibility for the registration of the submission.

4.5 Worked Example: StopPoints

A NaPTAN AnnotatedStopPoint instance is used to reference each of the vehicle stops where passengers may embark or disembark. Each of these identifies a NaPTAN point.

In the example there are nine stops, each with a specified type and sub type. See Table 4-2.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>StopPoint / Name</th>
<th>StopPoint / Name</th>
<th>AtcoCode</th>
<th>NaPTAN Code (SMS number)</th>
<th>Stop Type</th>
<th>Sub Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Bus Station</td>
<td>Suborn</td>
<td>0600000001</td>
<td>chsadad</td>
<td>BCS</td>
<td>MKD</td>
</tr>
<tr>
<td>#2</td>
<td>Shops</td>
<td>Garden Village</td>
<td>0600000002</td>
<td>chsadag</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#3</td>
<td>Plough</td>
<td>Robridge</td>
<td>0600000003</td>
<td>chsadaj</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#4</td>
<td>Red Lion</td>
<td>Barford</td>
<td>0600000004</td>
<td>chsadam</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#5</td>
<td>Golden Lion</td>
<td>Egham</td>
<td>0600000005</td>
<td>chsadap</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#6</td>
<td>Church</td>
<td>Godhill</td>
<td>0600000006</td>
<td>chsadat</td>
<td>BCT</td>
<td>CUS</td>
</tr>
<tr>
<td>#7</td>
<td>Exchange</td>
<td>Beall</td>
<td>0600000007</td>
<td>chsadaw</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#8</td>
<td>Shell</td>
<td>Beall</td>
<td>0600000008</td>
<td>chsadga</td>
<td>BCT</td>
<td>MKD</td>
</tr>
<tr>
<td>#9</td>
<td>Bus Station</td>
<td>Beall</td>
<td>0600000009</td>
<td>chsadgd</td>
<td>BCS</td>
<td>MKD</td>
</tr>
</tbody>
</table>

Table 4-2 – Worked Example: StopPoint Instances

4.6 Worked Example: Route and Tracks

A Route describes the sequence of stop points of the route, and contains an ordered collection of references to RouteSection elements. Each RouteSection comprises an ordered collection of RouteLink elements, making up the detailed stop sequence of the route. Links always run from NaPTAN StopPoint to NaPTAN StopPoint. The spatial path of each RouteLink is described by one or more Track elements, each having a Mapping; a collection of points (Location elements) giving the physical path of the route between stops.

Figure 4-1 shows an example route: The route links all have tracks comprising several location points.
A single RouteSection, with a link sequence of eight RouteLink instances (RL_01 – RL_08) suffices to connect the nine stops of the example, see Table 4-3. Each of the eight route links has a single Track, except for the link between ‘Garden Village’ and ‘Robridge, Plough’, which runs on two different roads (A1, B256), and so requires two track instance (T_2_1, T_2_2). The tracks have a varying number of intermediate points, depending on their spatial depiction.

<table>
<thead>
<tr>
<th>Link</th>
<th>Track</th>
<th>Origin</th>
<th>Destination</th>
<th>Mapping</th>
<th>Map Ref</th>
<th>Distance</th>
<th>Bearing</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL_01</td>
<td>T_1_1</td>
<td>Suborn, Bus Station</td>
<td>Garden Village, Shops</td>
<td>$(x_1,y_1), (x_2,y_2), (x_3,y_3), (x_4,y_4)$</td>
<td>A1</td>
<td>5573</td>
<td>E</td>
</tr>
<tr>
<td>RL_02</td>
<td>T_2_1</td>
<td>Garden Village, Shops</td>
<td>(function)</td>
<td>$(x_5,y_5), (x_6,y_6)$</td>
<td>A1</td>
<td>4512</td>
<td>NE</td>
</tr>
<tr>
<td></td>
<td>T_2_2</td>
<td>(function)</td>
<td>Robridge, Plough</td>
<td>$(x_7,y_7), (x_8,y_8)$</td>
<td>B256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL_03</td>
<td>T_3_1</td>
<td>Robridge, Plough</td>
<td>Barford, Red Lion</td>
<td>$(x_0,y_0), (x_9,y_9), (x_{10},y_{10})$</td>
<td>B256</td>
<td>6046</td>
<td>E</td>
</tr>
<tr>
<td>RL_04</td>
<td>T_4_1</td>
<td>Barford, Red Lion</td>
<td>Egham, Golden Lion</td>
<td>$(x_{11},y_{11}), (x_{12},y_{12})$</td>
<td>B256</td>
<td>2520</td>
<td>NE</td>
</tr>
<tr>
<td>RL_05</td>
<td>T_5_1</td>
<td>Egham, Golden Lion</td>
<td>Godhill, Church</td>
<td>$(x_{13},y_{13}), (x_{14},y_{14})$</td>
<td>B12</td>
<td>1955</td>
<td>SE</td>
</tr>
<tr>
<td>RL_06</td>
<td>T_6_1</td>
<td>Godhill, Church</td>
<td>Beall, Exchange</td>
<td>$(x_{15},y_{15}), (x_{16},y_{16})$</td>
<td>B12</td>
<td>2963</td>
<td>SW</td>
</tr>
<tr>
<td>RL_07</td>
<td>T_7_1</td>
<td>Beall, Exchange</td>
<td>Beall, Shell</td>
<td>$(x_{17},y_{17}), (x_{18},y_{18})$</td>
<td>B12</td>
<td>3560</td>
<td>SW</td>
</tr>
<tr>
<td>RL_08</td>
<td>T_8_1</td>
<td>Beall, Shell</td>
<td>Beall, Bus Station</td>
<td>$(x_{19},y_{19}), (x_{20},y_{20})$</td>
<td>B12</td>
<td>2005</td>
<td>SW</td>
</tr>
</tbody>
</table>
4.7 Worked Example: JourneyPattern

A JourneyPattern represents the pattern of working for vehicles of the service, and is composed of an ordered collection of JourneyPatternSection instances, each containing an ordered collection of JourneyPatternTimingLink instances, together defining a specific sequence of timing links.

To model the bus service in the worked example, one JourneyPattern instance is used that defines an overall sequence of stops.

- Three main vehicle journey instances (37, 38 and 40) are defined that reference the journey pattern; each vehicle journey omits particular stops (see Figure 4-2).
- Four other vehicle journey instances (39, 41, 42 and 43) are exact replicas of the first three vehicle journeys, apart from a different departure time, and so can be defined simply by referencing the links of the appropriate journey.

<table>
<thead>
<tr>
<th>Journey Pattern</th>
<th>Vehicle Journeys</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1. Suborn, Bus Station</td>
<td>#1. Suborn, Bus Station</td>
</tr>
<tr>
<td>#2. Garden Village, Shops</td>
<td>#2. Garden Village, Shops</td>
</tr>
<tr>
<td>#3. Robridge, Plough</td>
<td>#3. Robridge, Plough</td>
</tr>
<tr>
<td>#4. Barford, Red Lion</td>
<td>#4. Barford, Red Lion</td>
</tr>
<tr>
<td>#5. Egham, Golden Lion</td>
<td>#5. Egham, Golden Lion</td>
</tr>
<tr>
<td>#6. Godhill, Church</td>
<td>#6. Godhill, Church</td>
</tr>
<tr>
<td>#8. Beall, Shell</td>
<td>#8. Beall, Shell</td>
</tr>
<tr>
<td>#9. Beall, Bus Station</td>
<td>#9. Beall, Bus Station</td>
</tr>
</tbody>
</table>

Figure 4-2 – Worked Example: Journey Pattern

The exact sequence of stops is given by a JourneyPattern. The journey pattern will also specify information about the use of the stop (which may vary according to service), in particular: The JourneyPatternTimingLink instances for the example journey pattern are shown in Table 4-4.

(i) The TimingStatus of each stop used in the route. Stops may be deemed principal points or time information points, or both. The principal points must appear in a timetable, and so are mandatory for TransXChange, while other stop points are non-enforceable stops of the journey;

(ii) The Activity that takes place at each stop. For example, picking up or setting down passengers. This may need to be overridden on individual vehicle journeys.

<table>
<thead>
<tr>
<th>Link id</th>
<th>Run Time [sec]</th>
<th>Distance [m]</th>
<th>StopPoint Ref</th>
<th>Stop Name</th>
<th>Timing Status</th>
<th>Wait Time</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>JL_1#2</td>
<td>360</td>
<td>5.573</td>
<td>JL_1a</td>
<td>Suborn, Bus Station</td>
<td>PTP</td>
<td>0</td>
<td>PickUp</td>
</tr>
<tr>
<td>JL_1#2</td>
<td>360</td>
<td>5.573</td>
<td>JL_1b</td>
<td>Garden Village, Shops</td>
<td>PTP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_2#3</td>
<td>300</td>
<td>4.512</td>
<td>JL_2a</td>
<td>Garden Village, Shops</td>
<td>PTP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_2#3</td>
<td>300</td>
<td>4.512</td>
<td>JL_2b</td>
<td>Robridge, Plough</td>
<td>PTP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_3#4</td>
<td>120</td>
<td>6.046</td>
<td>JL_3a</td>
<td>Robridge, Plough</td>
<td>PTP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_3#4</td>
<td>120</td>
<td>6.046</td>
<td>JL_3b</td>
<td>Barford, Red Lion</td>
<td>PTP</td>
<td>60</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_4#5</td>
<td>180</td>
<td>2.520</td>
<td>JL_4a</td>
<td>Barford, Red Lion</td>
<td>PTP</td>
<td>60</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_4#5</td>
<td>180</td>
<td>2.520</td>
<td>JL_4b</td>
<td>Egham, Golden Lion</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_5#6</td>
<td>180</td>
<td>1.955</td>
<td>JL_5a</td>
<td>Egham, Golden Lion</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_5#6</td>
<td>180</td>
<td>1.955</td>
<td>JL_5b</td>
<td>Godhill, Church</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_5#6</td>
<td>180</td>
<td>1.955</td>
<td>JL_5c</td>
<td>Beall, Exchange</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_6#7</td>
<td>420</td>
<td>2.963</td>
<td>JL_6a</td>
<td>Godhill, Church</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_6#7</td>
<td>420</td>
<td>2.963</td>
<td>JL_6b</td>
<td>Beall, Exchange</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_7#8</td>
<td>60</td>
<td>3.560</td>
<td>JL_7a</td>
<td>Beall, Exchange</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_7#8</td>
<td>60</td>
<td>3.560</td>
<td>JL_7b</td>
<td>Beall, Shell</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
<tr>
<td>JL_7#8</td>
<td>60</td>
<td>3.560</td>
<td>JL_7a</td>
<td>Beall, Shell</td>
<td>TIP</td>
<td>0</td>
<td>PickUpAndSetDown</td>
</tr>
</tbody>
</table>
4.8 Worked Example: Line

Line elements are used to model the labelling of lines for the public. A service may have a number of lines, each with a LineName, and each vehicle journey can be assigned to a line. Normally, the same line is used to label vehicle journeys following the same pattern, but sometimes different journey variants with distinct patterns and link sequences may all be labelled under the same line name (though usually they will always have at least a few stops in common). Note that each VehicleJourney has a VehicleJourneyCode that is quite independent of the Line and LineName with which it may be associated.

In our example, there are three line names ‘1A’, ‘1B’, and ‘1C’, used to distinguish the different journeys that follow the three different journey patterns.

4.9 Worked Example: VehicleJourney

A VehicleJourney represents the traversal of a journey pattern at a particular time, and is composed of an ordered collection of VehicleJourneyTimingLink instances.

The timing links for VehicleJourney ‘40’ of the worked example are shown in Table 4-5. Two stops are skipped: ‘Robridge, Plough’ and ‘Beall Bus Exchange’.

<table>
<thead>
<tr>
<th>TimingLink Properties</th>
<th>VehicleJourneyStopUsage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Run Time [sec]</td>
</tr>
<tr>
<td>VL_1</td>
<td>360</td>
</tr>
<tr>
<td>VL_1</td>
<td>360</td>
</tr>
<tr>
<td>VL_1</td>
<td>360</td>
</tr>
<tr>
<td>VL_2</td>
<td>300</td>
</tr>
<tr>
<td>VL_3</td>
<td>120</td>
</tr>
<tr>
<td>VL_4</td>
<td>180</td>
</tr>
<tr>
<td>VL_4</td>
<td>180</td>
</tr>
<tr>
<td>VL_5</td>
<td>180</td>
</tr>
<tr>
<td>VL_6</td>
<td>180</td>
</tr>
<tr>
<td>VL_7</td>
<td>180</td>
</tr>
<tr>
<td>VL_7</td>
<td>180</td>
</tr>
</tbody>
</table>

Table 4-5 – Worked Example: Timing Links for VehicleJourney 1A

4.10 Worked Example: Operational Times

The operational times of the example are modelled as follows:

- There is a Service / ValidityPeriod from 5th February 2001 until further notice.
- The OperatingProfile / RegularDayType will show MondayToFriday operation.
- The OperatingProfile / BankHolidayOperation / DaysOfOperation has a Value of MayDay to show that it runs on Mayday.
- The OperatingProfile / BankHolidayOperation / DaysOfNonOperation has a Value of Christmas to show that it does not run on Christmas or Boxing Day.
5 EXAMPLES

5.1 Feature Examples

TransXChange is accompanied by a set of examples designed to illustrate the use of each of its features. For each example, a web page with links to the following is provided:

- Summary of features demonstrated by example.
- A route map.
- A matrix timetable representation.
- The XML encoding of the example.
- The TransXChange Publisher output of the encoded XML document.
- Explanatory notes describing the representation and implementation of specific features.

The examples can be found at http://www.transxchange.org.uk/examples.htm. There is a different set of examples for each release of TransXChange.

Table 5-1 lists the TransXChange examples, with the features covered by each case.

<table>
<thead>
<tr>
<th>Group</th>
<th>Name/Topology</th>
<th>Features covered</th>
<th>Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>Linear</td>
<td>• Linear route.</td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Registration details.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RouteTrack maps</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tracks, including instructions and Mapping System References.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Frequency based journey times, specified as an interval.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Registration Schema.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Inbound &amp; Outbound</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Operator Details (including TXC v2.4 attributes)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• AssistanceService (including TXC 2.5 attributes)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• VehicleAccessEquipment (including TXC 2.5 attributes)</td>
<td></td>
</tr>
<tr>
<td>Express</td>
<td>A linear route</td>
<td>• Express service.</td>
<td>General</td>
</tr>
<tr>
<td></td>
<td>with express</td>
<td>• Reuse of vehicle journey timing links in multiple journeys.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>journey patterns</td>
<td>• Overriding of Journey Pattern Run times,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Holiday day type exclusions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local stop point definitions for an off-street bus station: BCQ and BCS stop types.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Local stop area definitions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Variable bay allocation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supporting document.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Journey Footnotes, Marketing Name</td>
<td></td>
</tr>
<tr>
<td>Cancellation</td>
<td>(TXC 2.4)</td>
<td>• Basic Cancellation details</td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Registration Operation &amp; service only.</td>
<td></td>
</tr>
<tr>
<td>Complex</td>
<td>Interchange</td>
<td>• An Interchange.</td>
<td>Registration</td>
</tr>
<tr>
<td></td>
<td>Two patterns</td>
<td>• Linear route, with different stop visiting pattern at one end.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>run by two different operators presented as the same Service.</td>
<td>• Express stops.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>presented as the same Service.</td>
<td>• Frequency based journey times, specified as an interval.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All vehicle</td>
<td>• Combining operating days from service, journey pattern and vehicle journey level.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>journeys have the same timings.</td>
<td>• Serviced organisation &amp; school dates.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• More than one operator.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Timetable notes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Inward and outward timetables for the same service, using a single route.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Registration Schema.</td>
<td></td>
</tr>
<tr>
<td>Circular</td>
<td>A circular route.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Circular route.</td>
<td>• Reuse of route sections.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Reuse of route sections.</td>
<td>• Dead runs (positioning links).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Dead runs (positioning links).</td>
<td>• Partial traversal of a journey pattern.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Partial traversal of a journey pattern.</td>
<td>• Operator Garage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Operator Garage.</td>
<td>• AVL data - Vehicle type ticket machine, duty crew.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• AVL data - Vehicle type ticket machine, duty crew.</td>
<td>• WGS84.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• WGS84.</td>
<td>• Block (Running Board).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Block (Running Board).</td>
<td>• General Schema.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• General Schema.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cloverleaf</th>
<th>A cloverleaf route shape with three petals</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multiple routes composed of common route sections.</td>
<td>• Multiple journey patterns composed of common. Journey pattern sections.</td>
</tr>
<tr>
<td>• Multiple routes composed of common route sections.</td>
<td>• Dynamic destination display.</td>
</tr>
<tr>
<td>• Dynamic destination display.</td>
<td>• General Schema.</td>
</tr>
<tr>
<td>• General Schema.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lollipop</th>
<th>A lollipop&quot; shaped route, with two parallel branches.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Circular and parallel sections.</td>
<td>• Reuse of journey pattern sections.</td>
</tr>
<tr>
<td>• Reuse of journey pattern sections.</td>
<td>• Stop Sequence Numbers.</td>
</tr>
<tr>
<td>• Stop Sequence Numbers.</td>
<td>• Layover Point.</td>
</tr>
<tr>
<td>• Layover Point.</td>
<td>• Connecting services.</td>
</tr>
<tr>
<td>• Connecting services.</td>
<td>• General Schema.</td>
</tr>
<tr>
<td>• General Schema.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eye</th>
<th>An ‘eye’ shaped route, with two alternative branches.</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Multiple routes composed of common route sections.</td>
<td>• Multiple journey patterns composed of common. Journey pattern sections.</td>
</tr>
<tr>
<td>• Multiple routes composed of common route sections.</td>
<td>• Stop Sequence Numbers.</td>
</tr>
<tr>
<td>• Stop Sequence Numbers.</td>
<td>• Local stop point definitions.</td>
</tr>
<tr>
<td>• Local stop point definitions.</td>
<td>• Bilingual stop names &amp; schedule (Cymraeg).</td>
</tr>
<tr>
<td>• Bilingual stop names &amp; schedule (Cymraeg).</td>
<td>• Block (Running Board).</td>
</tr>
<tr>
<td>• Block (Running Board).</td>
<td>• Registration Schema.</td>
</tr>
<tr>
<td>• Registration Schema.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flexible</th>
<th>Use of flexible zones</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flexible zones.</td>
<td>• Flexible time bands.</td>
</tr>
<tr>
<td>• Flexible zones.</td>
<td>• Registration Schema.</td>
</tr>
<tr>
<td>• Flexible time bands.</td>
<td>• Registration Schema.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Footnotes</th>
<th>Use of footnotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flexible zones.</td>
<td>• Frequent Journeys</td>
</tr>
<tr>
<td>• Frequent Journeys</td>
<td>• Large number of services (144)</td>
</tr>
<tr>
<td>• Large number of services (144)</td>
<td>• Page overflow.</td>
</tr>
<tr>
<td>• Page overflow.</td>
<td>• Garage detail</td>
</tr>
<tr>
<td>• Garage detail</td>
<td>• Short Notice Registration</td>
</tr>
<tr>
<td>• Short Notice Registration</td>
<td>• Operational Data</td>
</tr>
<tr>
<td>• Operational Data</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hail &amp; Ride</th>
<th>Use of hail and ride stops</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Hail and ride sections.</td>
<td>• Local stop point definitions.</td>
</tr>
<tr>
<td>• Hail and ride sections.</td>
<td>• Full lollipop topology.</td>
</tr>
<tr>
<td>• Full lollipop topology.</td>
<td>• Frequency based journey times, specified as minutes past the hour (but not a Frequent Service).</td>
</tr>
<tr>
<td>• Frequency based journey times, specified as minutes past the hour (but not a Frequent Service).</td>
<td>• Short notice registration details.</td>
</tr>
<tr>
<td>• Short notice registration details.</td>
<td>• Scottish Bank Holidays.</td>
</tr>
<tr>
<td>• Scottish Bank Holidays.</td>
<td>• Workflow attributes</td>
</tr>
<tr>
<td>• Workflow attributes</td>
<td>• AccessibilityBooking (TXC v2.5)</td>
</tr>
<tr>
<td>• AccessibilityBooking (TXC v2.5)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Large Route</th>
<th>Very large timetable</th>
</tr>
</thead>
<tbody>
<tr>
<td>• More stops than fit on a page.</td>
<td>• More journeys than fit on a page.</td>
</tr>
<tr>
<td>• More stops than fit on a page.</td>
<td>• Registration</td>
</tr>
<tr>
<td>• More journeys than fit on a page.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Merge</th>
<th>Frequent Journeys</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Combing of individual journeys into a single column.</td>
<td>• Page overflow.</td>
</tr>
<tr>
<td>• Combing of individual journeys into a single column.</td>
<td>• Non PTP points.</td>
</tr>
<tr>
<td>• Page overflow.</td>
<td>• Use of default operating Profile.</td>
</tr>
<tr>
<td>• Non PTP points.</td>
<td>• Use of default operating Profile.</td>
</tr>
<tr>
<td>• Use of default operating Profile.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grouping (+TXC v2.4)</th>
<th>Journey Groupings</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Use of Journey Grouping (+TXC v2.4)</td>
<td>• Built-in Journey Grouping Labels. (+TXC v2.4)</td>
</tr>
<tr>
<td>• Built-in Journey Grouping Labels. (+TXC v2.4)</td>
<td>• Custom Journey Groupings. (+TXC v2.4)</td>
</tr>
<tr>
<td>• Custom Journey Groupings. (+TXC v2.4)</td>
<td>• Calendars (+TXC v2.4)</td>
</tr>
<tr>
<td>• Calendars (+TXC v2.4)</td>
<td>• Registration</td>
</tr>
<tr>
<td>• Registration</td>
<td>• Registration</td>
</tr>
<tr>
<td>• Registration</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registration</th>
<th>Goddess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goddess</td>
<td></td>
</tr>
</tbody>
</table>
2.5

5.2 Real Examples

A small number of ‘real’ examples, contributed by TransXChange users are also included on the web site. These provide larger examples using realistic data on an as is basis:

<table>
<thead>
<tr>
<th>Name</th>
<th>Features covered</th>
<th>TXC</th>
<th>Courtesy</th>
</tr>
</thead>
</table>
| U10_SOX_PH_U5_20090701                    | • Cancellation.  
• Multiple lines within the service; timetables marked with comments defining operating periods (university term/vacation).  
• Co-ordinates in WGS84. Daily variations. | 2.1 | Stagecoach |
| 3_V1_PF_3_20071028                        | • Lollipop " route;  
• Frequent interval covering part of length of route so described in an attachment.  
• Co-ordinates in WGS84. | 2.1 | Stagecoach |
| 7_SSS_PB_78_20080727                       | • Frequent intervals assigned to trips and also timetable described as frequency,  
• Holiday timetables,  
• Circular services,  
• Co-ordinates in WGS84. | 2.1 | Stagecoach |
| 86_STA_PD_R86_20070903                     | • Co-ordinates in Eastings/ Northings. | 2.1 | Stagecoach |
| S5_SOX_PH_S5_20090125.zip                  | • Complex service with route variations.  
• Monday-Friday timetable split into Monday-Wednesday/Thursday/Friday reflecting operational requirements.  
• Bank holiday operating profiles. | 2.1 | Stagecoach |
| X5_ST_PF_78S_20080426                      | • Limited stop service; attachment showing alternative routes (not affecting stopping places).  
• Co-ordinates in WGS84. | 2.1 | Stagecoach |

Figure 5-1 – Table List of TransXChange real examples
6.1 TransXChange Schema Overview

In a TransXChange document, data is organised around two main element types; Service (which may contain either or both StandardService or FlexibleService elements) and VehicleJourney, which together combine the instances of other elements into descriptions of bus schedules. Service instances are grouped under the TransXChange root element within a Services container, and VehicleJourney instances in a VehicleJourneys container. Other high-level elements such as Operator, Registration, RouteSection and JourneyPatternSection are also declared globally within containers under the TransXChange root element so that they may be reused in many different services (or even outside the context of a service, for general exchange).

The TransXChange element thus contains a number of different child and descendant elements (Figure 6-5) which can be characterised as falling into four groups:

- Topographical elements: StopPoint, StopArea, NptgLocality, ServicedOrganisation.
- Route and Network topology elements: Route, RouteSection, RouteLink.
- Registration Elements: Operator, Registration, (ShortNoticeRegistration).
- Ancillary elements: SupportingDocument. This can be specified at either the document or the registration level.
6.2 TransXChange Root Element

Every TransXChange document has a single instance of the TransXChange root element, which contains all the other elements.

- The TransXChange attributes specify document level attributes. See below.
- TransXChange Element Group: There are different Groups for the Registration & general schemas, reflecting slight different constraints on the registration & general elements.
- The PublishingOptions allows presentation options to be included. These can be described by a different embedded schema
- The Constraints specify referential integrity rules. See Later.

Table 6-1 – TransXChange Root element
6.2.1 TransXChange Element Attributes

The TransXChange element has the following attributes:

- Document Modification Attributes group
  - CreationDateTime: Timestamp of document creation date and time.
  - ModificationDateTime: Timestamp of document last modification date and time.
  - RevisionNumber: Monotonically increasing version number.
  - FileName: Name of file containing the document.
  - ChangesSince: If file contains only delta changes since a given date (TXC v2.4)

- Document Metadata Attributes group
  - SchemaVersion: TransXChange schema version identifier used for the document content model. Fixed: must be the schema version, e.g. ‘2.0’.
  - MappingSystem: Data system to use for mapping references (‘OS’, ‘Navtech’, etc) within the document.
  - DataSource: Name of provider of the data

- Fixed Attributes
  - LocationSystem: Data system to use for location coordinate references within the document: ‘WGS84’ or ‘Grid’. Must be ‘Grid’ for registration documents.
  - RegistrationDocument: Whether the document should be published as a registration, i.e. satisfy the additional semantic integrity constraints. Boolean.

6.2.2 TransXChange Root Attributes

Figure 6-3 – Attributes for TransXChange Schema
Figure 6-4 – TransXChange Change management attributes
6.2.3 TransXChange Child Elements

The **TransXChange** element contains a group element comprises the following child elements:

- **ServicedOrganisations**: A collection of **ServicedOrganisation** elements. See later.
- **NptgLocalities**: A collection of references to NPTG localities used in local stop definitions in the schedule. See later.
- **StopPoints**: A collection of the NaPTAN stop points used in the schedule. See later.
- **StopAreas**: A collection of reusable **StopArea** instances declared locally to group any stop points declared locally. See later.
- **RouteSections**: A collection of reusable **RouteSection** elements for defining routes. See later.
- **Routes**: A collection of reusable **Route** elements for use in journey patterns. See later.
- **JourneyPatternSections**: A collection of reusable **JourneyPatternSection** elements for defining journey patterns. See later.
- **Operators**: A collection of **Operator** elements. See later.
- **Services**: A collection of **Service** elements. See later.
- **VehicleJourneys**: A collection of **VehicleJourney** elements. See later.
- **Registrations**: A collection of **Registration** elements, each referencing a **Service** element. See later.
  - In the **TransXChange Registration Schema**, there must be one **Registration**.
  - In the **TransXChange General Schema** documents, there may be zero, one or many **Registration** instances.
- **SupportingDocuments**: A collection of reusable **SupportingDocument** elements. See later.
Figure 6-5 – Top Level Elements of TransXChange
6.3  Stop & Topographical Elements – StopPoints and Zones

6.3.1  NptgLocalities Element

The use of stops in TransXChange is based on NaPTAN. See the StopPoints element which allows stop usages to be declared. All stops are assigned to an NPTG Locality by means of a reference to a NPTG Locality identifier. When publishing the stop with a tool such as the TransXChange Publisher, the bus stop common names may be qualified with a locality name, for example “Barset, High Street”, rather than just “High Street”. It is therefore desirable that a TransXChange document contain the NPTG locality names so that a document can be published without recourse to the NPTG database.

For stops that are externally referenced (using an AnnotatedStopPointRef instance), the NptgLocality / LocalityName can be included as an annotation on the stop point reference. However for new stops that are defined locally using a StopPoint element, the locality names need to be supplied with a separate AnnotatedNptgLocalityRef, as they are not part of a new StopPoint declaration.

- NptgLocalities: A collection of AnnotatedNptgLocalityRef instances. See below.

```
<Sample NptgLocalities Element>
<TransXChange:NptgLocalitiesStructure>

<AnnotatedNptgLocalityRef>

  <NptgLocalityRef>12345</NptgLocalityRef>
  <LocalityName>Barset</LocalityName>
  <LocalityQualifier>Kent</LocalityQualifier>

</AnnotatedNptgLocalityRef>

<AnnotatedNptgLocalityRef>

  <NptgLocalityRef>67890</NptgLocalityRef>
  <LocalityName>High Street</LocalityName>
  <LocalityQualifier>London</LocalityQualifier>

</AnnotatedNptgLocalityRef>

</TransXChange:NptgLocalitiesStructure>
```

Figure 6-6 – NptgLocalities Element

6.3.2  AnnotatedNptgLocalityRef Element

Each AnnotatedNptgLocalityRef instance provides a local copy of NPTG Locality name information that can be used without recourse to the full NPTG database.

- NptgLocalityRef: Unique NPTG Locality identifier, i.e. NptgLocalityCode of NptgLocality
- LocalityName: Text name of NptgLocality; this name can be repeated locally so that the schedule may be annotated by tools such as the TransXChange Publisher without necessarily accessing the full NPTG database.
- LocalityQualifier: Any Qualifier of text name of locality, for example “Kent” to distinguish ‘Ashford (Kent)’ from ‘Ashford (Middlesex)’. 

6.3.3 StopPoints Element

The use of stops in TransXChange is based on NaPTAN. The StopPoints element (Figure 6-8) contains reusable declarations of the stops used by the routes and journey patterns of the schedule. All StopPointRef instances elsewhere in a document are resolved against the contents of the StopPoints element.

- Existing NaPTAN StopPoint instances can be referred to simply by using an AnnotatedStopPointRef to reference a NaPTAN system stop identifier – the AtcoCode of the stop. For further details refer to the NaPTAN Schema Guide.
- Stops may also be declared within a TransXChange XML document, by means of a local StopPoint declaration within the StopPoints container element. This can be used to provide full descriptive details of a new stop not yet in the NaPTAN database, or to correct details about an existing stop.

6.3.4 AnnotatedStopPointRef Element

The AnnotatedStopPoint element (Error! Reference source not found.) references an existing NaPTAN stop and comprises the following elements:

- StopPointRef: Unique NaPTAN identifier, i.e. AtcoCode of StopPoint.
- CommonName: Common text name of StopPoint; this name is repeated locally so that the schedule may be interpreted by tools such as the TransXChange Publisher without necessarily accessing the full NaPTAN database.
• **Indicator:** Further structured text descriptor element of **StopPoint**; that is used to distinguish similar stops, for example bus station bays.

• **LocalityName:** Text name of **NptgLocality**; this name can be repeated locally so that the schedule may be annotated by tools such as the TransXChange Publisher without necessarily accessing the full NPTG database.

• **LocalityQualifier:** Any Qualifier of text name of locality, for example “Kent” to distinguish ‘Ashford (Kent)’ from ‘Ashford (Middlesex).’

---

**Figure 6-9 – Annotated StopPointRef Element**

6.3.5 **StopPoint Element (Stop)**

The **StopPoint** element declares locally defined stops. A local **StopPoint** declaration uses **NaPTAN** elements, and must include a **NaPTAN** identifier for the stop. Local declarations are for expediency in cases when the **NaPTAN** definition for a new stop (or a change to an existing stop) has not yet been promulgated to the **NaPTAN** database. Even then, the **NaPTAN** identifier for such stops must be allocated by the relevant local transport authority. The other details of the stop may change subsequently in the course of registering it with the Authority.

Refer to the **NaPTAN 2.0 Schema Guide** for a definition of the **StopPoint** Element and its subelements.

6.3.6 **StopArea Element**

A **StopArea** is used to group stops: locally declared **StopPoint** instances can be assigned to one or more stop areas.

- **NaPTAN** stops that exist in the **NaPTAN** database may already have a **StopArea** element (previously called a **StopGroup**) associated with them.
- Local definitions of individual **StopArea** elements may also be declared within the **StopAreas** element of the **TransXChange** root element. Each **StopArea** must have a **StopAreaCode**. Local stop area declarations are for expediency in cases when the **NaPTAN** definition for a new stop area has not yet been promulgated to the **NaPTAN** database.
- Locally declared **StopPoint** elements may reference one or more **StopArea** instances.
- When importing schedules, an application will attempt to find the **StopArea** details in the **NaPTAN** database using the **StopAreaCode**. Only if no **StopArea** is found for the code will the locally supplied definition be used.
A NaPTAN StopArea is identified by an AreaCode, a unique NaPTAN identifier of the stop area.

Refer to the NaPTAN 2.0 Schema guide for a definition of the StopArea Element and its subelements.
6.4 Network Elements – Routes and Tracks

6.4.1 Route Element

A Route (Figure 6-10) describes the physical traversal of a bus along a route, described as an ordered collection of RouteLink elements, grouped into RouteSection elements. It is identified by a unique id attribute, and has the following properties:

- **PrivateCode**: an optional cross reference to an external system identifier for the route.
- **Description**: A textual description of the route.
- **RouteSectionRef**: An ordered collection of one or more references to RouteSection elements that contain the route links making up the route.
- **ReversingManoeuvre**: Used to describe any reversing manoeuvres needed.

Figure 6-10 – Route Element

6.4.2 RouteSection Element

A RouteSection (Figure 6-11) describes the course of a section of a route between several NaPTAN stops, and comprises an ordered collection of RouteLink elements, each describing a stop-to-stop path. A RouteSection can be used in multiple routes. It is identified by a unique id attribute.
6.4.3 RouteLink Element

A RouteLink (Figure 6-12) describes the course of a route between two NaPTAN stops. It is identified by a unique id attribute, and comprises:

- RouteRef: Optional reference to parent Route. Normally not stated as given by containing context, but may be specified when using the RouteLink as a stand-alone artefact. If already given by context, this value is ignored. (+TXC v2.4).
- From: The StopPointRef to the stop at which the link starts.
- To: The StopPointRef to the stop at which the link ends.
- Distance: The length of the path along the route in meters.
- Direction: Direction of the Route running over the RouteLink. See Table 6-2.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inbound</td>
<td>Inbound Direction.</td>
</tr>
<tr>
<td>outbound</td>
<td>Outbound Direction.</td>
</tr>
<tr>
<td>clockwise</td>
<td>Clockwise Direction.</td>
</tr>
<tr>
<td>antiClockwise</td>
<td>Anti-Clockwise Direction.</td>
</tr>
</tbody>
</table>

Table 6-2 – Allowed Values for Link / Direction

- Track: A description of the path of the link as one or more Track elements.
Figure 6-12 – RouteLink Element

A piece of the network topology connecting two stops.

- @id
- @CreationDateTime
- @ModificationDateTime
- @Modification
- @RevisionNumber

RouteLink

_@id_ 1, ∞

- RouteLinkStructure

RouteLinkStructure

- Attributes

  ParentRouteRef

  Type: RouteRefStructure

  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).

  From

  Type: ScpPointWRefStruct

  The stop at which the link begins.

  To

  Type: ScpPointWRefStruct

  The stop at which the link ends.

  Distance

  Type: DistanceType

  Distance in metres along track of the link.

  Direction

  Type: LineDirectionEnumeration

  Direction of the route running over the link. Enumerated value.

  Track

  Type: TrackStructure

  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 implementations. For example, a major road junction might represent the end of one track and the beginning of the next.

  Extensions

  Type: ExtensionsAnyStructure

  Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).
6.4.4 Route / Track Element

A Track element (Figure 6-13) describes the path of a route link between NaPTAN stops, and optionally, intermediate junction points. It comprises:

- **RouteLinkRef**: Optional reference to parent RouteLink. Normally not stated as given by containing context, but may be specified when using the RouteLink as a stand-alone artefact. If already given by context, this value is ignored. (+TXC v2.4).
- **Mapping**: A description of the path of the route as a series of geospatial points.
- **MapSystemReference**: An optional reference to an Ordnance Service TOID or other map feature identifier, using the mapping data system specified by the MappingSystem attribute.
- **Instructions**: Optional detailed step-by-step text instructions for navigating the track.

It is up to the implementer to choose the granularity of tracks – a given route might be represented by none, one, several or many tracks. Typically a track will be used for each distinct road or mapping layer feature that the implementer wishes to associate with part of the route.

Figure 6-13 – Track Element

6.4.5 Track Subelements

6.4.5.1 Track / Mapping Element

A Mapping element (Figure 6-14) describes the spatial path of a route link between NaPTAN stops that can be plotted on a map, as a series of at least two geospatial points: These points are independent of the stop point coordinates (though end points may reference the same coordinate) i.e. to plot a route the last and first point of each successive mapping will be connected.

- **Location**: A point in either WGS84 or grid coordinates. See Common Schema Elements later.
6.4.5.2 Track / Instructions Element

The *Instructions* element (Figure 6-15) provides an additional description of the path of a step of a route between *NaPTAN* stops as text instructions, and an ordered collection of structured elements:

- **Summary**: A free text description of the path of the route.
- **Feature**: A structured description of one or more steps of the journey.

### 6.4.5.3 Track / Instructions / Feature Element

The *Feature* element (Figure 6-15) describes a step of a route between *NaPTAN* stops:

- **LocationRef**: Reference to a *Location* in the *Track’s Mapping* instance that locates the feature on a map.
- **FeatureType**: Describes the type of feature encountered see Table 6-3.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>legOrigin</td>
<td>The start point of the leg.</td>
</tr>
<tr>
<td>legDestination</td>
<td>The end point of the leg.</td>
</tr>
<tr>
<td>bend</td>
<td>A bend in the track that merits attention (without a junction).</td>
</tr>
<tr>
<td>crossing</td>
<td>Cross over the road.</td>
</tr>
<tr>
<td>bridge</td>
<td>Traversing over a bridge.</td>
</tr>
<tr>
<td>junction</td>
<td>Either a point at which another road is taken, or a side road that is passed along the way.</td>
</tr>
<tr>
<td>miniRoundabout</td>
<td>Going round a small roundabout.</td>
</tr>
<tr>
<td>roadChange</td>
<td>Denotes a change of road name when there is no junction.</td>
</tr>
<tr>
<td>roundabout</td>
<td>Going round a small roundabout.</td>
</tr>
<tr>
<td>subway</td>
<td>Going through a subway.</td>
</tr>
<tr>
<td>trafficLights</td>
<td>Going through traffic lights.</td>
</tr>
<tr>
<td>landmark</td>
<td>A named landmark that can be seen from the track. The name should be provided in the Feature Description.</td>
</tr>
</tbody>
</table>

**Table 6-3 – Allowed Values for FeatureType**

- **RelativeBearing:** Which way you would turn from this feature to go to the next one. See Table 6-4.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>left</td>
<td>Left</td>
</tr>
<tr>
<td>right</td>
<td>Right</td>
</tr>
<tr>
<td>straightAhead</td>
<td>Straight ahead</td>
</tr>
<tr>
<td>uTurn</td>
<td>U-turn</td>
</tr>
</tbody>
</table>

**Table 6-4 – Allowed Values for RelativeBearing**

- **AbsoluteBearing:** The compass bearing which you should take directly from this feature point to go to the next one.
- **OnwardName:** The name of the road or path following this feature
- **RoadNumber:** The number of the road following this feature, e.g. ‘A1’.
- **Distance:** The distance to the next feature point, or to the leg alight point for the last feature point.
- **Description:** A text description of the individual feature.
6.5 Registration Elements: Operator, Registration, ShortNoticeRegistration

6.5.1 Operators Element

The Operators element (Figure 6-16) contains instances of the two different kinds of operator element that may be referenced by a Service:

- **Operator**: An operator definition allowing partial definition of an operator. See below.
- **LicensedOperator**: A full definition of an operator as is required for a registration.

In addition it can also be used to declare other types of organisation:

- **Contributor**: A holder of data rights other than an operator. See below.

The Operator and LicensedOperator elements differ only as which of their child elements are required or optional.

![Figure 6-16 – Operators Element]

![Figure 6-17 – Operators Element]

6.5.2 Operator Element

The Operator element (Figure 6-18) describes the Operator of a service. Every operator has an id attribute. References to operators within the document are made through the id (rather than the OperatorCode or the NationalOperatorCode), in order to guarantee a unique reference. Operator comprises:

- **NationalOperatorCode**: Unique national identifier of operator. This element is to support a future planned national operator code.
- **OperatorCode**: Unique Identifier of operator within document.
- **PrivateCode**: Alternative Identifier of operator for support of legacy codes. (+TXC v2.4.)
- **OperatorNamesGroup**: Contains elements relating to the Operator’s name. See below
- **OperatorLicenceGroup**: Contains elements relating to the Operator’s Licence. See below.
- **OtherLicences**: Alternative Licences for the operator. (+TXC v2.4)
- **OperatorParentGroup**: Information the relationship with other parent operators. See below. (+TXC v2.4)
- **OperatorInfoGroup**: Further information about the operator. See below. (+TXC v2.4)
- **OperatorContactGroup**: Information about how to contact the operator. See below. (+TXC v2.4)
- **AccessibilityBookings**: Information about making booking for wheelchair and accessible use for services which the operator runs. See below. (+TXC v2.5).
- **Garages**: The garages which the operator runs. See below.
- **ContributorGroup**: Information about data rights held by operator. See below. (+TXC v2.4).

---

**OperatorStructure**

- **Operator**
  - A transport operator.
  - `@id`, `@CreationDateTime`, `@ModificationDateTime`, `@Modification`, `@RevisionNumber`.

**Attributes**

- **NationalOperatorCode**
  - Unique national identifier of operator.
  - `Type`: `NationalOperatorCodeType`.

- **OperatorCode**
  - Identifier of operator.
  - `Type`: `OperatorCodeType`.

- **PrivateCode**
  - Alternative code for operator. TX v 2.4
  - `Type`: `PrivateCodeType`.

- **OperatorNamesGroup**
  - Elements for Operator. Names

- **OperatorLicenceGroup**
  - Elements for Operator primary Licence.

- **OtherLicences**
  - Other licences that the Operator holds (TXC 2.4).
  - `Type`: `OperatorLicencesStructure`.

- **OperatorParentGroup**
  - Elements for Operator Parents.
  - **ParentOperatorRef**
    - Immediate Parent of Operator. TXC v 2.4
    - `Type`: `AnnotatedOperatorRefStructure`.
  - **UltimateParentRef**
    - Ultimate Parent of Operator. TXC v 2.4
    - `Type`: `AnnotatedOperatorRefStructure`.

- **OperatorInfoGroup**
  - Elements for Operator further description (TXC v 2.3).

- **OperatorContactGroup**
  - Contact details for Operator.

- **AccessibilityBookings**
  - Information about accessibility booking (+TXC v2.5).
  - `Type`: `AccessibilityBookingsStructure`.

- **Garages**
  - Bus garages that operator runs.
  - `Type`: `GaragesStructure`.

- **ContributorGroup**
  - Elements that characterise a contributor. (TXC v 2.4)
  - `Type`: `ContributorGroup`.

- **Extensions**
  - Extensions to schema. (Wrapper tag used to avoid problems with handling of optional "any" by some validators).
  - `Type`: `ExtensionsAnyStructure`.
6.5.3 LicensedOperator Element

The LicensedOperator element (Figure 6-18) is identical to the Operator element except that certain fields are mandatory.

- OperatorNameOnLicence, LicenceNumber, LicenceClassification.
- LicensedOperatorContactGroup: ContactTelephoneNumber, EnquiryTelephoneNumber, OperatorAddresses.
LicensedOperatorStructure
A transport operator that includes licensing details that are mandatory for service registration.
* @id,  
  * @CreationDateTime,  
  * @ModificationDateTime,  
  * @Modification,  
  * @RevisionNumber.

**LicensedOperator**

**attributes**

- **Unique national identifier of operator.**
  - **NationalOperatorCode** (type NationalOperatorCodeType)

- **Identifier of operator.**
  - **OperatorCode** (type OperatorCodeType)

- **Alternative code for operator. TX v 2.4**
  - **PrivateCode** (type PrivateCodeType)

- **Short text name for operator.**
  - **OperatorShortName** (type OperatorShortNameType)

- **Full name of the operator (i.e. as appears on licence). @lang. Mandatory for a Licenced Operator**
  - **OperatorNameOnLicence** (type NaturalLanguageStringStructure)

- **Name under which operator trades.**
  - **TradingName** (type OperatorTradingNameType)

- **Name under which operator is referenced**
  - **ReferenceName** (type OperatorTradingNameType)

- **Operator’s licence number.**
  - **LicenceNumber** (type OperatorLicenceNumberType)

- **Type of operator licence. Enumerated value.**
  - **LicenceClassification** (type OperatorLicenceEnumeration)

- **Date of expiry of licence. TXC v 2.4**
  - **LicenceExpiryDate** (type xsd:date)

- **Names of licence holders TXC v 2.4**
  - **LicenceHolderNames** (type LicenceHolderNamesStructure)

- **Status of the Operator’s licence, Default is valid. TXC v 2.4**
  - **LicenceStatus** (type LicenceStatusEnumeration)

**OtherLicences**

**type** OperatorLicencesStructure

**Extensions**

**type** ExtensionsAnyStructure

**Figure 6-19 – LicensedOperator Element**
6.5.4 Operator & LicensedOperator: Subelements

6.5.4.1 Operator / OperatorNamesGroup

The **OperatorNamesGroup** (Figure 6-20) describes the various names of an **Operator** and comprises:

- **OperatorShortName**: Short text name for operator.
- **OperatorNameOnLicence**: Full name of the operator, as it appears on licence.
- **TradingName**: The name under which operator trades.
- **ReferenceName**: Name used to distinguish operator from similarly named operators. For use by data providers and managers. Not normally for public use. (+TXC v2.4).

![OperatorNamesGroup Diagram]

**Figure 6-20 – OperatorNamesGroup**

6.5.4.2 Operator / OperatorLicenceGroup

The **OperatorLicenceGroup** (Figure 6-21) describes the licence details for an **Operator** of a service and comprises:

- **LicenceNumber**: Operator's licence number.
- **LicenceClassification**: Type of licence that the operator has. See Table 6-5.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>standardNational</td>
<td>Standard National Licence type.</td>
</tr>
<tr>
<td>standardInternational</td>
<td>Standard International Licence type.</td>
</tr>
<tr>
<td>restricted</td>
<td>Restricted Licence type.</td>
</tr>
<tr>
<td>specialRestricted</td>
<td>Special Restricted Licence type.</td>
</tr>
<tr>
<td>communityBusPermit</td>
<td>Community Bus Permit Licence type.</td>
</tr>
</tbody>
</table>

**Table 6-5 – Allowed Values for LicenceClassification**

- **LicenceExpiryDate**: Date of Expiry of Operator's licence. (+TXC v2.4.)
- **LicenceHolderNames**: Names listed as licence holders. (+TXC v2.4.)
- **LicenceStatus**: Status of Operator's licence. (+TXC v2.4.)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid</td>
<td>Valid</td>
</tr>
<tr>
<td>Refused</td>
<td>Refused</td>
</tr>
<tr>
<td>Surrendered</td>
<td>Surrendered</td>
</tr>
<tr>
<td>Continuation not sought</td>
<td>Continuation not sought</td>
</tr>
<tr>
<td>Revoked</td>
<td>Revoked</td>
</tr>
<tr>
<td>Withdrawn</td>
<td>Withdrawn</td>
</tr>
<tr>
<td>Application in progress.</td>
<td>Application in progress.</td>
</tr>
</tbody>
</table>
Table 6-6 – Allowed Values for LicenceStatus

<table>
<thead>
<tr>
<th>LicenceNumber</th>
<th>type</th>
<th>OperatorLicenceNumberType</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operator's licence number.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LicenceClassification</th>
<th>type</th>
<th>OperatorLicenceEnumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of operator licence.</td>
<td>Enumerated</td>
<td>value.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LicenceExpiryDate</th>
<th>type</th>
<th>xsd:date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date of expiry of licence.</td>
<td>TXC v2.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LicenceHolderNames</th>
<th>type</th>
<th>LicenceHolderNamesStructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Names of licence holders.</td>
<td>TXC v2.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LicenceStatus</th>
<th>type</th>
<th>LicenceStatusEnumeration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status of the Operator's licence, Default is valid. TXC v2.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-21 – Operator / OperatorLicenceGroup

6.5.4.3 Operator / OtherLicences Element (+TXC v2.4)

The **Operator / OtherLicences** (Figure 6-22) element additional licences that the **Operator** holders:

- **OtherLicences**: Collection of Licences. (+TXC v2.4)
  - **OperatorLicence**: See OperatorLicenceGroup for details. (+TXC v2.4)

Figure 6-22 – Operator / OperatorLicence Element

6.5.4.4 Operator / OperatorParentGroup (+TXC v2.4)

The **OperatorParentGroup** (Figure 6-21) describes the relationships of an **Operator** with any parent operator and comprises:

- **ParentOperatorRef**: Immediate parent of Operator. (+TXC v2.4)
  - **OperatorRef**: Identifier of Operator.
  - **OperatorNamesGroup**: See above

Elements for Operator. Parents.

OperatorParentGroup

AnnotatedOperatorRefStructure

Immediate Parent of Operator. TXC v2.4

ParentOperatorRef

Reference to an operator

OperatorRef

Unique national identifier of operator.

NationalOperatorRef

NationalOperatorCodeType

Ultimate Parent of Operator. TXC v2.4

UltimateParentRef

AnnotatedOperatorRefStructure

Elements for Operator. Names

OperatorNamesGroup

Short text name for operator.

OperatorShortName

OperatorShortNameType

Full name of the operator (i.e. as appears on licence).

OperatorNameOnLicence

NaturalLanguageStringStructure

Operator Public Name. Name under which operator trades.

TradingName

OperatorTradingNameType

Name used to distinguish operator from similarly named operators. For use by data providers and managers, not normally for public use. (TXC v2.4)

ReferenceName

xsd:normalizedString

Figure 6-23 – Operator / OperatorParentGroup

6.5.4.5 Operator / OperatorInfoGroup (+TXC v2.4)

The OperatorInfoGroup (Figure 6-24) describes further details for an Operator of a service and comprises:

- **PrimaryMode**: The main mode the operator provides, e.g. bus, coach, etc. (+TXC v2.4).
- **EbsrUser**: Whether Operator provides timetables to VOSA Electronic Bus Service Registration (EBSR). Default is true. (+TXC v2.4).
- **TravelineOwner**: Primary Traveline Region who owns data record. (+TXC v2.4). Table 6-7 shows the allowed values.
- **RegionalOperatorRefs**: Translation of operator codes to other alias for different regions. (+TXC v2.4). See below.
- **Note**: Further comment on the operator (+TXC v2.4).

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW</td>
<td>South West</td>
</tr>
<tr>
<td>WA</td>
<td>Wales</td>
</tr>
<tr>
<td>YO</td>
<td>Yorkshire</td>
</tr>
<tr>
<td>XS</td>
<td>MDV combined area for London &amp; South East</td>
</tr>
<tr>
<td>WM</td>
<td>West Midlands</td>
</tr>
<tr>
<td>SC</td>
<td>Scotland</td>
</tr>
<tr>
<td>NW</td>
<td>North West</td>
</tr>
<tr>
<td>NE</td>
<td>North East</td>
</tr>
<tr>
<td>Other</td>
<td>Other</td>
</tr>
</tbody>
</table>
### Table 6-7 – Allowed Values for Data Owner

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>PrimaryMode</em></td>
<td><code>VehicleModesEnumeration</code></td>
<td>Primary mode that operator runs.</td>
</tr>
<tr>
<td><em>EbsrUser</em></td>
<td><code>xsd:boolean</code></td>
<td>Whether operator provides timetables to EBSR. Default is true.</td>
</tr>
<tr>
<td>TravelineOwner</td>
<td><code>RegionShortCodeEnumeration</code></td>
<td>Current Traveline owner of data.</td>
</tr>
<tr>
<td>RegionalOperatorRefs</td>
<td><code>RegionOperatorCodeStructure</code></td>
<td>Mapping to legacy regional codes.</td>
</tr>
<tr>
<td>Note</td>
<td><code>NaturalLanguageStringStructure</code></td>
<td>Further comment on operator.</td>
</tr>
</tbody>
</table>

#### Figure 6-24 – Operator / OperatorInfoGroup

6.5.4.6 Operator / RegionalOperatorRef (+TXC v2.4)

The **RegionalOperatorRef** (Figure 6-26) maps a regional operator code to the operator:
- **TravelineRegion**: Traveline Region in which code is used (+TXC v2.4). Table 6-7 above shows the allowed values.
- **TravelineOperatorCode**: Code used in specified Traveline Region.

#### Figure 6-25 – Operator / RegionalOperatorRef

6.5.4.7 Operator / OperatorContactGroup

The **OperatorContactGroup** (Figure 6-26) describes the contact details for an **Operator** of a service and comprises:
- **EnquiryTelephoneNumber**: Telephone Number for public enquiries to the operator concerning the service. See *TelephoneContactStructure* in common schema elements in Section 7.
- **ContactTelephoneNumber**: Telephone Number to contact operator concerning the service. See *TelephoneContactStructure* below.
- **ContactFaxNumber**: Fax Number to contact operator concerning the service. See *TelephoneContactStructure* below.
- **ContactPerson**: Name of contact person.
- **CustomerServiceTelephoneNumber**: General customer Service Telephone Number to contact operator See *TelephoneContactStructure* below.
• **OperatorAddresses**: Operator's addresses. A separate **OperatorAddress** and **CorrespondanceAddress** can be specified. See **PostalAddressStructure** in Common Schema Elements in Section 7.

• **EmailAddress**: The email address of the operator. It is up to the operator whether an individual's address or a generic company e-mail address is used.

• **WebSiteAddress**: The web site URL of the operator.

---

**Figure 6-26 – Operator / OperatorContactGroup**

The **AccessibilityBooking** element records details about how to book wheelchair use of an operator's services (Figure 6-29).

- **Mode**: Transport mode of operator’s service to which this information relates.
- **Regions**: Travel line regions to which to which this information relates for an operator.
  - **RegionRef** reference to a Traveline region.
- **AssistanceAvailability**: Availability of services for wheelchair users (Table 6-8). shows the allowed values for assistance availability.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Assistance service is not available from Operator.</td>
</tr>
<tr>
<td>available</td>
<td>Assistance is available from Operator.</td>
</tr>
<tr>
<td>availableIfBooked</td>
<td>Assistance is available if booked.</td>
</tr>
<tr>
<td>availableAtCertainTimes</td>
<td>Assistance is available at certain times.</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known if available.</td>
</tr>
</tbody>
</table>

**Table 6-8 – Allowed Values for AssistanceAvailability**
- **ContactTelephoneNumber**: Telephone Number to contact operator concerning accessibility. See *TelephoneContactStructure* below.
- **EmailAddress**: Email address to contact operator concerning accessibility.
- **BookingMethod**: Methods of booking allowed. See Table 6-12 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>Book Online</td>
</tr>
<tr>
<td>callDriver</td>
<td>Book by phoning driver</td>
</tr>
<tr>
<td>callOffice</td>
<td>Book by phoning Office</td>
</tr>
<tr>
<td>phonesAtStop</td>
<td>Book using phone at stop</td>
</tr>
<tr>
<td>text</td>
<td>Book by sending text</td>
</tr>
<tr>
<td>none</td>
<td>No method</td>
</tr>
<tr>
<td>other</td>
<td>Other Booking Method</td>
</tr>
</tbody>
</table>

Table 6-9 – Allowed Values for BookingMethod

- **MinimumBookingPeriod**: Minimum interval in advance of departure day or time that service may be ordered.
- **LatestBookingTime**: Latest time in the day that a booking can be made.
- **BookingUrl**: The web site address for accessibility info and or booking.
- **Note**: Additional note on booking arrangements.
6.5.4.9 Operator / Garages Element

The Operator / Garages element records details about the garages or depots which the operator uses. It contains a collection of Garage (Figure 6-28) elements. Each Garage is composed of:

- **GarageCode**: Identifier of garage. This will be referenced by other elements.
- **GarageName**: Name of garage.
- **ContactNumber**: Telephone Number to contact for queries about operational data. See TelephoneContactStructure in Common Schema Elements in Section 7.
- **Address**: Postal Address of garage. See PostalAddressStructure in Common Schema Elements in Section 7.
- **Location**: Spatial coordinates of garage.

![Diagram of Garage Element]

Figure 6-28 – Operator / Garages / Garage Element

6.5.4.10 Operator / ContributorGroup (+TXC v2.4)

The ContributorGroup element records details about the data rights management for the operator. (Figure 6-29):

- **PolicyStatus**: Status under the Freedom of Information Act. (TXC v2.4). See Table 6-10 for allowed values. Defaults to ExemptFromFreedomOfInformation. If not specified, use value of any containing context.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubjectToFreedomOfInformation</td>
<td>Holder is Subject to the Freedom Of Information Act</td>
</tr>
<tr>
<td>ExemptFromFreedomOfInformation</td>
<td>Holder is not Subject to the Freedom Of Information Act</td>
</tr>
<tr>
<td>Other</td>
<td>Other status</td>
</tr>
</tbody>
</table>

Table 6-10 – Allowed Values for Policy Status

- **ContributorClassification**: Type of Contributor. (TXC v2.4). See Table 6-11 for allowed values.
Table 6-11 – Allowed Values for Contributor Classification

- **DataRights**: One or more Data rights offered by Organisation. See below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommunityMember</td>
<td>Contributor is a private individual</td>
</tr>
<tr>
<td>ProfessionalOrganisation</td>
<td>Contributor is a professional Organisation</td>
</tr>
<tr>
<td>LocalAuthority</td>
<td>Contributor is a local authority.</td>
</tr>
<tr>
<td>ProfessionalOrganisationForLocalAuthority</td>
<td>Contributor is a professional Organisation acting on behalf of a Local Authority.</td>
</tr>
</tbody>
</table>

Figure 6-29 – Operator / ContributorGroup Element

6.5.4.11 DataRight / (+TXC v2.4)

The **DataRight** element records details about a data rights (Figure 6-29):

- **TermsOfUse** elements:
- **AllowedUse**: Permitted use of data element or elements. (TXC v2.4). See Table 6-12 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommercialLicence</td>
<td>Data may be used under a commercial licence</td>
</tr>
<tr>
<td>OpenLicence</td>
<td>Data may be used under an Open Source Licence</td>
</tr>
<tr>
<td>NotForProfit</td>
<td>Data has may be used for free under an Open Source Licence provided service does not itself charge user for s use.</td>
</tr>
<tr>
<td>Unrestricted</td>
<td>Data may be used without charge, subject to terms and conditions.</td>
</tr>
</tbody>
</table>

Table 6-12 – Allowed Values for AllowedUse

- **Copyright** elements:
  - **CopyrightUrl**: Reference to web URL with copyright statement. (TXC v2.4).
  - **CopyrightStatement**: Text for copyright. (TXC v2.4).

- **Policy** elements:
- **PolicyStatus**: Policy status of element covered by right. (TXC v2.4). See Table 6-13 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SubjectToFreedomOfInformation</td>
<td>Element is Subject to the Freedom Of information Act</td>
</tr>
<tr>
<td>ExemptFromFreedomOfInformation</td>
<td>Element is not Subject to the Freedom Of information Act</td>
</tr>
<tr>
<td>Other</td>
<td>Other status</td>
</tr>
</tbody>
</table>

Table 6-13 – Allowed Values for PolicyStatus

- **PolicyUrl**: URL providing information on the policy. (TXC v2.4).
- **PolicyJustification**: Text providing information on the policy (TXC v2.4).

- **DataRights**: One or more Data rights offered by Organisation. See below.
6.5.5 Registration Element

The **Registration** element (Figure 6-31) records statutory administrative details about the registration of the service. In the **TransXChange Registration Schema** the element is mandatory; in the **TransXChange General Schema** it is not. A **Registration** comprises:

- **ServiceRef**: The **Service** that the registration covers.
- **RegistrationSubmissionGroup**: Describes basic properties of registration.
- **RegistrationInfoGroup**: Describes further properties of the registration.
- **ShortNoticeRegistration**: Additional information to support a registration made with less than the statutory period of notice. See later below.
RegistrationStructure (extension)

A registration of a service.
- @id,
- @CreationDateTime,
- @ModificationDateTime,
- @Modification,
- @RevisionNumber.

RegistrationSubmissionGroup

Submission information about the registration.

RegistrationInfoGroup

Information about the registration.

ShortNoticeRegistration

Additional information to support a short notice registration.

Extensions

Extensions to schema. (Wrapper tag used to avoid problems with handling of optional ‘any’ by some validators).

Figure 6-31 – Registration Element

6.5.6 Registration / RegistrationSubmissionGroup

The RegistrationSubmissionGroup (Figure 6-32) holds elements describing the basic submission of registration.

- **SubmissionDate**: Intended date of Registration submission by submitter (officially received date may be different).
- **VosaRegistrationNumber**: The identifiers for the Registration. See below.
- **RegistrationWorkflowGroup**: Elements concerning the processing of the Registration. See below.
- **VariationNumber**: Variation number of the registration.
- **SubmissionAuthor**: Contact details of person submitting registration. See below.
- **EbsrAgent**: Name of Agency if registration is prepared and submitted by a proxy
- **TrafficAreas**: A collection of TrafficArea instances with full or partial responsibility for the registration of the submission. See below.
- **CirculatedAuthorities**: Collection of CirculatedAuthority instances to whom the registration is to be circulated. See below.
6.5.7 Registration / RegistrationWorkflowGroup

The **RegistrationWorkflowGroup** (Figure 6-32) holds elements describing the basic submission of registration.

- **ApplicationClassification**: Type of the registration application. See Table 6-14.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>New registration.</td>
</tr>
<tr>
<td>chargeableChange</td>
<td>Chargeable modification of an existing registration.</td>
</tr>
<tr>
<td>nonChargeableChange</td>
<td>Non-chargeable modification of an existing registration.</td>
</tr>
<tr>
<td>cancel</td>
<td>Cancellation of a registration.</td>
</tr>
</tbody>
</table>

Table 6-14 – Allowed Values for Registration / ApplicationClassification

- **RegistrationWorkflowStatus**: Current processing status of the application. See Table 6-15. Allows tracking of current status of a submission. (+TXC 2.4)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>draft</td>
<td>Document is a working preapproval draft</td>
</tr>
<tr>
<td>submitted</td>
<td>Document is a submitted schedule not yet approved</td>
</tr>
<tr>
<td>rejected</td>
<td>Document has been rejected</td>
</tr>
<tr>
<td>elaboratedAfterAcceptance</td>
<td>Document is an accepted schedule that has been further augmented with details.</td>
</tr>
<tr>
<td>underConsultation</td>
<td>Document is a submitted proposal that has been acknowledged by VOSA but is not yet approved</td>
</tr>
</tbody>
</table>
accepted | Document is an accepted schedule  
variant | Document is an issued variant that does not need approval  
other | Document has some other status – e.g. for a non-registration  

Table 6-15 – Allowed Values for Registration / RegistrationWorkflowStatus

- **RegistrationVariantNature**: Nature of non-statutory variant effect. Indicates nature of change i.e. how document will affect downstream systems. See Table 6-16. (+TXC 2.4)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Document changes all or most aspects of registration</td>
</tr>
<tr>
<td>journeys</td>
<td>Document has changes to add or remove journeys that will affect schedule</td>
</tr>
<tr>
<td>journeyPattern</td>
<td>Document has changes to alter stops of journeys that will affect schedule</td>
</tr>
<tr>
<td>other</td>
<td>Change is not ascribed to any section</td>
</tr>
<tr>
<td>operational</td>
<td>Document has changes to operational data that does not affect schedule</td>
</tr>
<tr>
<td>profile</td>
<td>Document has changes to availability that may affect schedule</td>
</tr>
<tr>
<td>registration</td>
<td>Document has changes to registration particulars</td>
</tr>
<tr>
<td>textual</td>
<td>Document has only textual changes</td>
</tr>
<tr>
<td>timings</td>
<td>Document has changes to timings of journeys that will affect schedule</td>
</tr>
<tr>
<td>track</td>
<td>Document has changes to route track data that does not affect schedule</td>
</tr>
</tbody>
</table>

Table 6-16 – Allowed Values for Registration / RegistrationVariantNature

- **RegistrationSubVariantNumber**: Subnumber identifying Non-statutory change to a previously submitted registration. Should be unique. (+TXC 2.4).
- **StatutoryChange**: Whether change affects formal registration. (+TXC 2.4). Default is true.

---

**Figure 6-33 – Registration Workflow Group**

6.5.8 Registration / RegistrationInfoGroup

The **RegistrationInfoGroup** (Figure 6-34) holds elements describing additional properties of a registration.

- **SubsidyDetails**: Information about any subsidy of the Service. See below.
• **ContractedService**: Information about any contract under which the *Service* is run for an authority. See below.

• **QualityPartnership**: Information about any Statutory Quality partnership under which the *Service* is run.

• **SupportingDocuments**: Names of additional documents that accompany the registration. Note that references to any schematic maps that are in image format should be placed with the *Service / SchematicMap* element, and not here.

```
Figure 6-34 – RegistrationInfoGroup
```

6.5.9 Registration Subelements

6.5.9.1 Registration / VosaRegistrationNumber Element

The *VosaRegistrationNumber* element (*Figure 6-35*) specifies the unique identifiers of the *Registration*. It is made up of three components:

• **TanCode**: Two character Traffic Area prefix. See *Table 6-17*.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>North Eastern Traffic Area</td>
</tr>
<tr>
<td>PC</td>
<td>North Western Traffic Area</td>
</tr>
<tr>
<td>PD</td>
<td>West Midlands Traffic Area</td>
</tr>
<tr>
<td>PF</td>
<td>Eastern Traffic Area</td>
</tr>
<tr>
<td>PG</td>
<td>Welsh Traffic Area</td>
</tr>
<tr>
<td>PH</td>
<td>Western Traffic Area</td>
</tr>
<tr>
<td>PK</td>
<td>South Eastern and Metropolitan Traffic Area</td>
</tr>
<tr>
<td>PM</td>
<td>Scottish Traffic Area</td>
</tr>
</tbody>
</table>

*Table 6-17 – Allowed Values for TanCode*

• **LicenceNumber**: The Registered operator’s seven character licence number. This should be the same as the *Operator / LicenceNumber* value.

• **RegistrationNumber**: Unique identifier of registration for licence holder. 1-4 numeric only characters.

When displayed, numbers include a separator slash between the licence number and the suffix, for example ‘PB1235601/456’.

6.5.9.2 Registration / SubmissionAuthor Element

The SubmissionAuthor (Figure 6-36) describes the signatory of the submission – that is, upon whose authority the submission is made. It comprises:

- **Position**: Position of the signatory of the Registration.
- **Title**: Title of the signatory of the Registration.
- **Forename**: Forename of the signatory of the Registration.
- **Surname**: Surname of the signatory of the Registration.

6.5.9.3 Registration / TrafficArea Element

The TrafficAreas element (Figure 6-37) lists the individual TrafficArea elements for the registration.

- **TrafficAreaName**: Specifies a TrafficArea – see Table 6-18.
Table 6-18 – Allowed Values for TrafficArea / Names

<table>
<thead>
<tr>
<th>TrafficArea</th>
<th>Value</th>
<th>TXC version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NorthEastern</td>
<td>Aberdeen</td>
<td>2.0</td>
</tr>
<tr>
<td>NorthWestern</td>
<td>Aberdeen</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthEastMetropolitan</td>
<td>Angus</td>
<td>2.0</td>
</tr>
<tr>
<td>Scottish</td>
<td>ArgyllAndBute</td>
<td>2.0</td>
</tr>
<tr>
<td>Welsh</td>
<td>BathAndNorthEastSomerset</td>
<td>2.0</td>
</tr>
<tr>
<td>Western</td>
<td>Bedford</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>CentralBedfordshire</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>Berkshire</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>BlackburnWithDarwen</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Blackpool</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>BlaenauGwent</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Bournemouth</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>BracknellForest</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Bridgend</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>BrightonAndHove</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Bristol</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Buckinghamshire</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Caerphilly</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Cambridgeshire</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Figure 6-37 – Registration / TrafficArea Element

6.5.9.4 Registration / CirculatedAuthorities Element

The CirculatedAuthorities element (Figure 6-38) lists the individual CirculatedAuthority elements for the registration.

- **CirculatedAuthority**: Names identifying circulated authority. May be specified in one of two ways:
  - **AuthorityName**: Name of circulated authority from validated list. Should be used if Authority is in current list. See Table 6-19.
  - **UnverifiedAuthorityName**: Unvalidated name of circulated authority. For other use. (TXC v2.4). Definitive name for new areas should be obtained from VOSA.
<table>
<thead>
<tr>
<th>Location</th>
<th>Description</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardiff</td>
<td>Cardiff</td>
<td>2.0</td>
</tr>
<tr>
<td>Carmarthenshire</td>
<td>Carmarthenshire</td>
<td>2.0</td>
</tr>
<tr>
<td>CentroWestMidlands</td>
<td>Centro (West Midlands)</td>
<td>2.0</td>
</tr>
<tr>
<td>Ceredigion</td>
<td>Ceredigion</td>
<td>2.0</td>
</tr>
<tr>
<td>Channel Islands</td>
<td>Channel Islands</td>
<td>2.0</td>
</tr>
<tr>
<td>Cheshire</td>
<td>Cheshire</td>
<td>2.0</td>
</tr>
<tr>
<td>CheshireEast</td>
<td>Cheshire East</td>
<td>2.2</td>
</tr>
<tr>
<td>CheshireWestAndChester</td>
<td>Cheshire West and Chester</td>
<td>2.2</td>
</tr>
<tr>
<td>Clackmannashire</td>
<td>Clackmannashire</td>
<td>2.0</td>
</tr>
<tr>
<td>ComhairleNanEileanSiar</td>
<td>Comhairle Nan Eilean Siar</td>
<td>2.0</td>
</tr>
<tr>
<td>Conwy</td>
<td>Conwy</td>
<td>2.0</td>
</tr>
<tr>
<td>CornwallAndScillies</td>
<td>Cornwall and Scillies</td>
<td>2.0</td>
</tr>
<tr>
<td>Cumbria</td>
<td>Cumbria</td>
<td>2.0</td>
</tr>
<tr>
<td>Darlington</td>
<td>Darlington</td>
<td>2.0</td>
</tr>
<tr>
<td>Denbighshire</td>
<td>Denbighshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Derby</td>
<td>Derby</td>
<td>2.0</td>
</tr>
<tr>
<td>Derbyshire</td>
<td>Derbyshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Devon</td>
<td>Devon</td>
<td>2.0</td>
</tr>
<tr>
<td>Dorset</td>
<td>Dorset</td>
<td>2.0</td>
</tr>
<tr>
<td>DumfriesAndGalloway</td>
<td>Dumfries and Galloway</td>
<td>2.0</td>
</tr>
<tr>
<td>Dundee</td>
<td>Dundee</td>
<td>2.0</td>
</tr>
<tr>
<td>Durham</td>
<td>Durham</td>
<td>2.0</td>
</tr>
<tr>
<td>EastAyrshire</td>
<td>East Ayrshire</td>
<td>2.0</td>
</tr>
<tr>
<td>EastDunbartonshire</td>
<td>East Dunbartonshire</td>
<td>2.0</td>
</tr>
<tr>
<td>EastLothian</td>
<td>East Lothian</td>
<td>2.0</td>
</tr>
<tr>
<td>EastRenfrewshire</td>
<td>East Renfrewshire</td>
<td>2.0</td>
</tr>
<tr>
<td>EastRidingOfYorkshire</td>
<td>East Riding of Yorkshire</td>
<td>2.0</td>
</tr>
<tr>
<td>EastSussex</td>
<td>East Sussex</td>
<td>2.0</td>
</tr>
<tr>
<td>Edinburgh</td>
<td>Edinburgh</td>
<td>2.0</td>
</tr>
<tr>
<td>Essex</td>
<td>Essex</td>
<td>2.0</td>
</tr>
<tr>
<td>ExeterCity</td>
<td>Exeter City</td>
<td>2.4</td>
</tr>
<tr>
<td>Falkirk</td>
<td>Falkirk</td>
<td>2.0</td>
</tr>
<tr>
<td>Fife</td>
<td>Fife</td>
<td>2.0</td>
</tr>
<tr>
<td>Flintshire</td>
<td>Flintshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Glasgow</td>
<td>Glasgow</td>
<td>2.0</td>
</tr>
<tr>
<td>Gloucestershire</td>
<td>Gloucestershire</td>
<td>2.0</td>
</tr>
<tr>
<td>GMPTE</td>
<td>GMPTE (Manchester)</td>
<td>2.0</td>
</tr>
<tr>
<td>Gwynedd</td>
<td>Gwynedd</td>
<td>2.0</td>
</tr>
<tr>
<td>Halton</td>
<td>Halton</td>
<td>2.0</td>
</tr>
<tr>
<td>Hampshire</td>
<td>Hampshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Hartlepool</td>
<td>Hartlepool</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Examples</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Havering</strong></td>
<td>Havering 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Herefordshire</strong></td>
<td>Herefordshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Hertfordshire</strong></td>
<td>Hertfordshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Highland</strong></td>
<td>Highland 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Inverclyde</strong></td>
<td>Inverclyde 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>IsleOfAnglesey</strong></td>
<td>Isle of Anglesey 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>IsleOfMan</strong></td>
<td>Isle of Man 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>IsleOfWight</strong></td>
<td>Isle of Wight 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Kent</strong></td>
<td>Kent 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>KingstonUponHull</strong></td>
<td>Kingston Upon Hull 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Lancashire</strong></td>
<td>Lancashire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Leicester</strong></td>
<td>Leicester 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Leicestershire</strong></td>
<td>Leicestershire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Lincolnshire</strong></td>
<td>Lincolnshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>London</strong></td>
<td>London 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Luton</strong></td>
<td>Luton 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Medway</strong></td>
<td>Medway 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Merseytravel</strong></td>
<td>Merseytravel 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>MerthyrTydfil</strong></td>
<td>Merthyr Tydfil 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>MetroWestYorks</strong></td>
<td>Metro (West Yorks) 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Middlesbrough</strong></td>
<td>Middlesbrough 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Midlothian</strong></td>
<td>Midlothian 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>MiltonKeynes</strong></td>
<td>Milton Keynes 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Monmouthshire</strong></td>
<td>Monmouthshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Moray</strong></td>
<td>Moray 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NeathPortTalbot</strong></td>
<td>Neath Port Talbot 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>WestBerkshire</strong></td>
<td>West Berkshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Newport</strong></td>
<td>Newport 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NexusTyneside</strong></td>
<td>Nexus (Tyneside) 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Norfolk</strong></td>
<td>Norfolk 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthAyrshire</strong></td>
<td>North Ayrshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthEastLincolnshire</strong></td>
<td>North East Lincolnshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthernIreland</strong></td>
<td>Northern Ireland 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthLanarkshire</strong></td>
<td>North Lanarkshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthLincolnshire</strong></td>
<td>North Lincolnshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthSomerset</strong></td>
<td>North Somerset 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorthYorkshire</strong></td>
<td>North Yorkshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Northamptonshire</strong></td>
<td>Northamptonshire 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Northumberland</strong></td>
<td>Northumberland 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>NorwichCity</strong></td>
<td>NorwichCity 2.4</td>
<td></td>
</tr>
<tr>
<td><strong>Nottingham</strong></td>
<td>Nottingham 2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Nottinghamshire</strong></td>
<td>Nottinghamshire 2.0</td>
<td></td>
</tr>
<tr>
<td>Region</td>
<td>Name</td>
<td>Version</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td>---------</td>
</tr>
<tr>
<td>OrkneyIslands</td>
<td>Orkney Islands</td>
<td>2.0</td>
</tr>
<tr>
<td>Oxfordshire</td>
<td>Oxfordshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Pembrokeshire</td>
<td>Pembrokeshire</td>
<td>2.0</td>
</tr>
<tr>
<td>PerthAndKinross</td>
<td>Perth and Kinross</td>
<td>2.0</td>
</tr>
<tr>
<td>Peterborough</td>
<td>Peterborough</td>
<td>2.0</td>
</tr>
<tr>
<td>Plymouth</td>
<td>Plymouth</td>
<td>2.0</td>
</tr>
<tr>
<td>Poole</td>
<td>Poole</td>
<td>2.0</td>
</tr>
<tr>
<td>Portsmouth</td>
<td>Portsmouth</td>
<td>2.0</td>
</tr>
<tr>
<td>Powys</td>
<td>Powys</td>
<td>2.0</td>
</tr>
<tr>
<td>Reading</td>
<td>Reading</td>
<td>2.0</td>
</tr>
<tr>
<td>RedcarAndCleveland</td>
<td>Redcar and Cleveland</td>
<td>2.0</td>
</tr>
<tr>
<td>Renfrewshire</td>
<td>Renfrewshire</td>
<td>2.0</td>
</tr>
<tr>
<td>RhonddaCynonTaff</td>
<td>Rhondda Cynon Taff</td>
<td>2.0</td>
</tr>
<tr>
<td>Rutland</td>
<td>Rutland</td>
<td>2.0</td>
</tr>
<tr>
<td>ScottishBorders</td>
<td>Scottish Borders</td>
<td>2.0</td>
</tr>
<tr>
<td>ShetlandIslands</td>
<td>Shetland Islands</td>
<td>2.0</td>
</tr>
<tr>
<td>Shropshire</td>
<td>Shropshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Slough</td>
<td>Slough</td>
<td>2.0</td>
</tr>
<tr>
<td>Somerset</td>
<td>Somerset</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthAyrshire</td>
<td>South Ayrshire</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthGloucestershire</td>
<td>South Gloucestershire</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthLanarkshire</td>
<td>South Lanarkshire</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthYorkshirePTE</td>
<td>South Yorkshire PTE</td>
<td>2.0</td>
</tr>
<tr>
<td>Southampton</td>
<td>Southampton</td>
<td>2.0</td>
</tr>
<tr>
<td>SouthendOnSea</td>
<td>Southend On Sea</td>
<td>2.0</td>
</tr>
<tr>
<td>Staffordshire</td>
<td>Staffordshire</td>
<td>2.0</td>
</tr>
<tr>
<td>Stirling</td>
<td>Stirling</td>
<td>2.0</td>
</tr>
<tr>
<td>StocktonOnTees</td>
<td>Stockton On Tees</td>
<td>2.0</td>
</tr>
<tr>
<td>StokeOnTrent</td>
<td>Stoke On Trent</td>
<td>2.0</td>
</tr>
<tr>
<td>StrathclydePTE</td>
<td>Strathclyde PTE</td>
<td>2.0</td>
</tr>
<tr>
<td>Suffolk</td>
<td>Suffolk</td>
<td>2.0</td>
</tr>
<tr>
<td>Surrey</td>
<td>Surrey</td>
<td>2.0</td>
</tr>
<tr>
<td>Swansea</td>
<td>Swansea</td>
<td>2.0</td>
</tr>
<tr>
<td>Swindon</td>
<td>Swindon</td>
<td>2.0</td>
</tr>
<tr>
<td>TelfordAndWrekin</td>
<td>Telford and Wrekin</td>
<td>2.0</td>
</tr>
<tr>
<td>Thurrock</td>
<td>Thurrock</td>
<td>2.0</td>
</tr>
<tr>
<td>Torbay</td>
<td>Torbay</td>
<td>2.0</td>
</tr>
<tr>
<td>Torfaen</td>
<td>Torfaen</td>
<td>2.0</td>
</tr>
<tr>
<td>ValeOfGlamorgan</td>
<td>Vale of Glamorgan</td>
<td>2.0</td>
</tr>
<tr>
<td>Warrington</td>
<td>Warrington</td>
<td>2.0</td>
</tr>
<tr>
<td>Warwickshire</td>
<td>Warwickshire</td>
<td>2.0</td>
</tr>
<tr>
<td>WestDunbartonshire</td>
<td>West Dunbartonshire</td>
<td>2.0</td>
</tr>
</tbody>
</table>
West Lothian | West Lothian | 2.0
West Sussex | West Sussex | 2.0
Wiltshire | Wiltshire | 2.0
Windsor and Maidenhead | Windsor and Maidenhead | 2.0
Wokingham | Wokingham | 2.0
Worcestershire | Worcestershire | 2.0
York | York | 2.0

Table 6-19 – Allowed Values for CirculatedAuthority Names

The **SubsidyDetails** element (Figure 6-39) gives information about any subsidy that applies to the Registration.

Either there are none – NoSubsidy, or there is a Subsidy, made up of two elements:

- **SubsidyType**: Whether subsidy is full or partial. **Table 6-20**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>partial</td>
<td>Partial subsidy applies.</td>
</tr>
<tr>
<td>full</td>
<td>Full subsidy applies.</td>
</tr>
</tbody>
</table>

**Table 6-20 – Allowed Values for SubsidyType**

- **SubsidisingAuthority**: Name of subsidising authority.

The **ContractedService** element (Figure 6-40) specifies if the service is run under contract to a Local Authority or SPT. This item is specific to Scottish registration.

Nature of Contract:

- **NotContracted**: Service is not run under contract.
- **WhollyContracted**: Service is run wholly under contract.
- **PartContracted**: Service is run in part under contract.
- **ContractingAuthority:** Names of one or more authorities awarding contract. See *CirculatedAuthority / AuthorityName*.

**Figure 6-40 – Registration / ContractedService Element**

6.5.9.7 Registration / SupportingDocument Element

The *SupportingDocument* element *(Figure 6-41)* Associates any supporting documents associated with the registration. Documents are identified by a *DocumentUri*.

**Figure 6-41 – Registration / SupportingDocument Element**
6.5.10 ShortNoticeRegistration Element

A short notice registration is an application to register, cancel or change a service made with less than the normally 56 days' period of notice. Only certain determined cases can be submitted within the reduced period. A ShortNoticeRegistration requires additional details as specified by one or more elements in the ChangeImpactGroup & ChangeJustificationGroup.

- **ChangeImpactGroup:** Elements describing the impact of the change.
- **ChangeJustificationGroup:** Elements describing the justification(s) for the change.

6.5.11 ShortNoticeRegistration / ChangeImpactGroup

The ChangeImpactGroup (Figure 6-43) holds elements describing the impact of the change. These include:

- **PublicAvailability:** Whether the service is to be available to the general public. See below.
- **ChangeImpact:** Whether the change to the service time is in excess of the normal allowed limits and so requires additional justification. See below.

6.5.12 ShortNoticeRegistration / ChangeJustificationGroup

The ChangeJustificationGroup (Figure 6-44) holds elements describing the justification(s) for the change. These include:

- **BankHolidayChange:** Whether the ShortNoticeRegistration is needed to address a bank holiday requirement. See below.
- **ChangeToConnectAlteredService:** Whether the short notice registration is needed to handle a modification to another service. See below.
- **ReplaceDiscontinuedService:** Whether the service is to replace a discontinued service, whose discontinuation justifies the short notice registration? See below.
- **LocalHolidayChange:** Whether the short notice registration is to accommodate a local holiday. See below.
- **SpecialOccasion:** Whether the short notice registration is to accommodate a special occasion. See below.
- **RegulationOrderCompliance:** Whether the short notice registration is needed to meet a road traffic order. See below.
- **ChangeRequestedByExternalAuthority:** Whether the short notice registration is needed to meet a request by an external authority such as the Police. See below.
- **ExceptionalRequirement:** Whether the short notice registration is needed to meet an allowed exceptional requirement. See below.
- **MiscellaneousJustification**: The reasons justifying the short notice registration submission where none of the above considerations are applicable. More than one reason may be included.

```xml
<ChangeJustificationGroup>
  <BankHolidayChange type="xsd:boolean">The ShortNoticeRegistration is to address a bank holiday requirement.</BankHolidayChange>
  <ChangeToConnectAlteredService type="ChangeToConnectAlteredServiceStructure">The change is to accommodate a change in a connecting service.</ChangeToConnectAlteredService>
  <ReplaceDiscontinuedService type="ReplaceDiscontinuedServiceStructure">The registration is to replace another service that has been discontinued.</ReplaceDiscontinuedService>
  <LocalHolidayChange type="LocalHolidayChangeStructure">The change is to accommodate a local holiday.</LocalHolidayChange>
  <SpecialOccasion type="SpecialOccasionStructure">The change is to allow for a special occasion.</SpecialOccasion>
  <RegulationOrderCompliance type="RegulationOrderComplianceStructure">The change is to comply with a traffic order note.</RegulationOrderCompliance>
  <ChangeRequestedByExternalAuthority type="ChangeRequestedByExternalAuthorityStructure">The change is to comply with a request from an external authority such as the Police.</ChangeRequestedByExternalAuthority>
  <ExceptionalRequirement type="ExceptionalRequirementStructure">The change is to accommodate an exceptional requirement.</ExceptionalRequirement>
  <MiscellaneousJustification type="NaturalLanguageStringStructure">Reasons justifying short notice registration submission where no predefined ShortNoticeRegistration reasons are applicable. @lang.</MiscellaneousJustification>
</ChangeJustificationGroup>
```

**Figure 6-44 – ShortNoticeRegistration / ChangeJustificationGroup**

### 6.5.13 ShortNoticeRegistration Subelements

#### 6.5.13.1 ShortNoticeRegistration / Public Availability Element

The **PublicAvailability** element (Figure 6-45) specifies whether the service is to be available to the general public.

- **AvailableToPublic**: Specifies service is available.
- **NotAvailableToPublic**: Specifies service is not available, accompanied by a **NonAvailabilityDescription**.
6.5.13.2 ShortNoticeRegistration / ChangeImpact Element

The ChangeImpact element (Figure 6-46) specifies whether the change to the service time is in excess of the normal allowed limit (i.e. more than ten minutes from the current time): if the change is more than the allowed amount, then a justification must be given, otherwise a Minor Change Description can be used.

- **ChangeExceedsLimit**: Change exceeds the allowed limit. Only possible if change to existing application, i.e. if ChangeClassification is Change or Cancel.
- **ChangeDoesNotExceedLimit**: The change does not exceed the limit.

6.5.13.3 ShortNoticeRegistration / ChangeToConnectAlteredService Element

The ChangeToConnectAlteredService (Figure 6-47) specifies whether the short notice registration is needed to handle a modification to another service, and if so, which one:

- **ServiceRef**: Reference to another Service definition provided elsewhere in the document.
- **Description**: Text description of the service &/or its identifier if not defined by a service reference.

6.5.13.4 ShortNoticeRegistration / ReplaceDiscontinuedService Element

The ReplaceDiscontinuedService (Figure 6-48) identifies the discontinued service which the service of the short notice registration replaces.

- **DiscontinuedServiceOperator**: Operator of the discontinued service.
- **DiscontinuedService**: Description of the discontinued service, an AnnotatedServiceRefStructure.
- **ServiceRef**: Reference to another Service definition provided elsewhere in the document.
- **Description**: Text description of the service &/or its identifier if not defined by a service reference.
Figure 6-48 – ShortNoticeRegistration / ReplaceDiscontinuedService Element

6.5.13.5 ShortNoticeRegistration / LocalHolidayChange Element

The **LocalHolidayChange** element (Figure 6-49) identifies the local holiday which justifies the short notice registration.

- **LocalHolidayNote**: Description of local holiday.

Figure 6-49 – ShortNoticeRegistration / LocalHolidayChange Element

6.5.13.6 ShortNoticeRegistration / SpecialOccasion Element

The **SpecialOccasion** element (Figure 6-50) identifies the special occasion which justifies the short notice registration.

- **SpecialOccasionName**: Name of special occasion.

Figure 6-50 – ShortNoticeRegistration / SpecialOccasion Element

6.5.13.7 ShortNoticeRegistration / RegulationOrderCompliance Element

The **RegulationOrderCompliance** element (Figure 6-51) identifies whether the short notice registration is to comply with a regulation order.

- **TrafficOrderNote**: Identifies the order.

Figure 6-51 – ShortNoticeRegistration / RegulationOrderCompliance Element
6.5.13.8 ShortNoticeRegistration / ChangeRequestedByExternalAuthority Element

The `ChangeRequestedByExternalAuthority` (Figure 6-52) specifies whether the short notice registration is needed to meet a request by an external authority such as the Police, and any explanation or corroboration of the change.

- **ChangeRequestDescription**: Explanation or corroboration of why the change is required.

![Diagram 6-52](image)

**Figure 6-52 – ShortNoticeRegistration / ChangeRequestedByExternalAuthority Element**

6.5.13.9 ShortNoticeRegistration / ExceptionalRequirement Element

The `ExceptionalRequirement` element (Figure 6-53) specified whether the registration is needed to meet an allowed exceptional requirement.

- **ChangeRequestDescription**: Explanation or corroboration of why the change is required.

![Diagram 6-53](image)

**Figure 6-53 – ShortNoticeRegistration / ExceptionalRequirement Element**
6.6 Service Description Elements

6.6.1 Services Element

Definitions of each Service describing a bus schedule are contained within the Services container element:

- In a TransXChange Registration schema document, only one registered service may be described at a time. The registered Service must reference a Registration, and the referenced Registration must describe the operator in full with a LicensedOperator. Instance. Relevant details of other connecting services may be included in the document as separate service declarations.
- In a TransXChange General Schema document, many services can be described.

6.6.2 Service Element

The Service element (Figure 6-54) describes a service. The elements include:

- ServiceCode: The unique identifier for the Service. This is of type string (+TXC v 2.4) to allow arbitrary characters to be used. Note that is using an Ampersand (&) the ‘&’ will have to be escaped, for example ‘S2&S2’ must be coded &lt;ServiceCode&gt;S1&amp;S2&lt;/ServiceCode&gt;
- PrivateCode: An identifier for the Service that can be used to associate it with other systems.
- Lines: The public identifiers for the Service. See later.
- OperatingPeriod: Period within which Service operates. See below.
- OperatingProfile: Default operational days for journeys running the Service. See Operational Days elements later.
- JourneyGroupings: Journey Groupings for the Service. See Below.
- ServiceClassification: Type of the Service. See below.
- ServiceOperationalGroup: Default operational elements associated with the Service. See below.
- RegisteredOperatorRef: Registered operator of the Service. See LicensedOperator and Operator. On a Registration Service this must reference a LicensedOperator instance.
- AssociatedOperatorRef: Another operator associated with the service in a secondary capacity. See Operator and LicensedOperator.
- ServiceInfoGroup: Further informational elements about the Service. See below.
- ServiceDescriptionGroup: Further descriptive elements about the service. See below.
- ServiceComponentGroup: Information about the routes and journeys patterns comprising the Service. See below.
ServiceStructure (extension)

- ServiceCode
  - type: ServiceCodeType
  - Code that uniquely identifies the Service.

- PrivateCode
  - type: PrivateCodeType
  - A private code that uniquely identifies the service. May be used for interoperating with other information systems.

Lines
  - type
  - The lines that make up the service.

OperatingPeriod
  - type: ServicePeriodStructure
  - Inclusive period within which Service runs.

OperatingProfile
  - type: OperatingProfileStructure
  - Pattern of normal and special days defining the operational profile of the service. Default is Monday to Friday, every day of the year.

JourneyGroupings
  - type: JourneyGroupingsStructure
  - Any subgroupings of journeys used within a service. Provides a means of collating and grouping journeys for presentation. (TXC v2.4)

ServiceClassification
  - type: ServiceClassificationStructure
  - Nature of Service, categorisation for statutory purpose.

ServiceOperationalGroup
  - Operational elements for service. Specifies defaults that may be overridden at some other levels.

RegisteredOperatorRef
  - type: OperatorRefStructure
  - Registered Operator of Service - i.e. with responsibility for the service. For registrations this must refer to a licensed operator.

AssociatedOperators
  - type: OperatorRoleStructure
  - 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
  - type: ExtensionsAnyStructure
  - Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 6-54 – Service Element
6.6.3 Service / ServiceInfoGroup

The ServiceInfoGroup (Figure 6-55) group holds informational elements describing the Service.

- **ServiceHasMirror**: Whether service has a corresponding service in the return direction.
- **StopRequirements**: Whether the service requires new stop declarations. See below.
- **Mode**: Transport mode of service. See Table 6-21. Default is bus. The equivalent NeTEx modes are shown in the right-hand column.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>NeTEx</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>Air service.</td>
<td>air</td>
</tr>
<tr>
<td>bus</td>
<td>Bus service.</td>
<td>bus</td>
</tr>
<tr>
<td>coach</td>
<td>Coach service.</td>
<td>coach</td>
</tr>
<tr>
<td>ferry</td>
<td>Ferry service.</td>
<td>water *</td>
</tr>
<tr>
<td>metro</td>
<td>Metro service.</td>
<td>metro</td>
</tr>
<tr>
<td>telecabeine</td>
<td>Lift or cableway service</td>
<td>cableway *</td>
</tr>
<tr>
<td>train</td>
<td>Train service.</td>
<td>train</td>
</tr>
<tr>
<td>tram</td>
<td>Tram service.</td>
<td>tram</td>
</tr>
<tr>
<td>underground</td>
<td>Underground service</td>
<td>metro</td>
</tr>
</tbody>
</table>

Table 6-21 – Allowed Values for Service / Mode

- **PublicUse**: Whether service allows public use, i.e. is not ‘Closed Door’.
- **ServiceAvailability**: Whether service has a corresponding service in the return direction. See below.
- **Express**: Whether service is flagged as an express (i.e. limited stop) service.
- **CommercialBasis**: on which the service is offered. May be overridden for specific journey patterns, vehicle Journeys and timing links. See Table 6-22 for allowed values. (+TXC v2.4)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contracted</td>
<td>Service on Link is contracted</td>
</tr>
<tr>
<td>notContracted</td>
<td>Service on Link is not contracted.</td>
</tr>
<tr>
<td>partContacted</td>
<td>Service on Link is part contracted.</td>
</tr>
<tr>
<td>unknown</td>
<td>Basis is unknown.</td>
</tr>
</tbody>
</table>

Table 6-22 – Allowed Values for Service / CommercialBasis
6.6.4 Service / ServiceDescriptionGroup

The ServiceDescriptionGroup (Figure 6-56) group holds ancillary descriptive elements describing the Service.

- **Description**: Text description of the services. On registrations should include “A description of the service or change for Notices & Proceedings”. For example, “a regular service at half-hourly intervals daytime on Mondays to Saturdays, and hourly in the evenings and on Sundays”.

- **Note**: Structured notes associated with service. See common schema elements later.

- **SchematicMap**: Name of any schematic map associated with services. File name. Must be an image file (png, gif, jpeg). Schematic maps must be provided for Registrations.

- **MarketingName**: Name to use when displaying service in some applications. (+TXC v2.4)
ToBeMarketedWith: Information on marketing of the services. See below.

- Description
  type: NaturalLanguageStringStructure
  Description of the service. @lang.

- Note
  type: NoteStructure
  0..
  Descriptive note about the service.

- SchematicMap
  type: ImageDocumentType
  0..
  File name of document containing a schematic map of the route. Must be a .png, .gif or .jpeg format document.

- MarketingName
  type: xsd:normalizedString
  Name used for marketing and in some display applications TXC v2.4

- ToBeMarketedWith
  type: ToBeMarketedWithStructure
  Information about other services with which the registered service will be jointly marketed.

Figure 6-56 – Service / ServiceDescriptionGroup

6.6.5 Service / ServiceComponentGroup

The ServiceComponentGroup (Figure 6-57) holds the fundamental timetable components of the Service.

- StandardService: Any standard service component.
- FlexibleService: Any flexible service component.
- Direction: The direction of the Service. See Table 6-2

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inbound</td>
<td>Inbound Direction.</td>
</tr>
<tr>
<td>outbound</td>
<td>Outbound Direction.</td>
</tr>
<tr>
<td>inboundAndOutbound</td>
<td>Inbound and Outbound Direction.</td>
</tr>
<tr>
<td>circular</td>
<td>Circular Direction.</td>
</tr>
<tr>
<td>clockwise</td>
<td>Clockwise Direction.</td>
</tr>
<tr>
<td>antiClockwise</td>
<td>Anti-Clockwise Direction.</td>
</tr>
</tbody>
</table>

Table 6-23 – Allowed Values for Service / Direction

- JourneyPatternInterchange: Zero or more interchanges at which the journey patterns of the service connect.
The *Line* element (*Figure 6-58*) allows one or more public identifiers of the service to be associated with the vehicle journeys of the service. For example, lines ‘1’, ‘1a’, ‘1b’. Each individual *VehicleJourney* element specifies the line or line variant that the journey runs. A *Line* provides an arbitrary label for presentational and marketing purposes and does not necessarily correspond to the strict route variants: the same line name may be used on services with different stopping patterns. A *Line* is identified by a unique *id* attribute.

Each *Line* has:
- **LineName**: Name of line, typically a number or letters & number
- **MarketingName**: Alternative marketing name for Line (+TXC v2.4).
- **Outbound Description**: Description of outbound direction of line for publicity purposes. (+TXC v2.4). See below
- **Inbound Description**: Description of inbound direction of line for publicity purposes. (+TXC v2.4). See below
- **LineFontColourGroup**: (+TXC v2.4). Optional preferred colour to use for when showing line labels in visual media. See below.
- **LineImage**: Url of an optional image associated with line to use in graphic media. (+TXC v2.4)
6.6.6.2 Line / Description Element (+TXC v2.4)

The LineDescription element (Figure 6-58) describes a direction of a line for publicity purposes

- **Origin.** Origin of line. (+TXC v2.4).
- **Destination.** Destination of line. (+TXC v2.4).
- **Vias.** Via points. (+TXC v2.4).
  - **Via.** Via point on line. (+TXC v2.4).
- **Description.** Description of line. (+TXC v2.4).
The **LineColourGroup** (Figure 6-58) specifies colour preferences for presenting the line consistently in media.

- **LineColour**: Optional preferred colour to use when showing line in graphic media. (+TXC v2.4)
- **LineFontColour**: Optional preferred colour to use for text when showing line labels in visual media. (+TXC v2.4)
- **AlternativeLineColour**: Second choice colour to use when showing line in graphic media. (+TXC v2.4)
- **LineColour**: Second choice colour to use when showing Line labels in visual media. (+TXC v2.4)
6.6.4 Service / OperatingPeriod Element

The OperatingPeriod element (Figure 6-61) states the period over which the Service operates. It includes:

- A StartDate: Date at which service commences.
- An EndDate: Date at which service ends. If absent, continues indefinitely.
- Recommended EndDate: Even if service is notionally open ended for registration purposes, there may be a recommended date after which not to use the data. (TXC v2.4)

See also OperationProfile element for further elements relating to the operating days of a service.

Figure 6-61 – Service / OperatingPeriod Element

6.6.5 Service / ServiceClassification Element

The ServiceClassification element (Figure 6-62) classifies the service as being one or more of a number of categories of service. The classifications are as follows:

- NormalStopping: A service where all stops on a route are used.
- LimitedStops: A service where only certain pre-defined stops on a route are used.
- HailAndRide: A service that stops anywhere on designated parts of the route, if flagged down by passengers where it is safe to do so.
- Flexible: A service running in accordance with the rules for a flexible service, with designated pickup and set down zones or points. Must be specified if service is a FlexibleService.
- ExcursionOrTour: A service where all passengers go to the same destination and return to their departure point. Further qualified by:
  - MaxDepartures: Maximum number of vehicle departures within one day associated with an excursion type service.
- RuralService: A service primarily aimed at serving rural communities (i.e. at locations with populations less than 25,000 people).
- SchoolOrWorks: A service dedicated to a school or works that is not available to the public.
- OtherService: Services that do not fit any of the defined categories. Should only be used sparingly:
  - Further explained by a Description.
Service types may be combined in any way. Normal combinations of service are shown in Table 6-24:

<table>
<thead>
<tr>
<th>Group</th>
<th>ServiceClassification</th>
<th>NormalStopping</th>
<th>LimitedStops</th>
<th>HailAndRide</th>
<th>FlexibleService</th>
<th>ExcursionOrTour</th>
<th>OtherService</th>
<th>SchoolOrWorks</th>
<th>RuralService</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>NormalStopping</td>
<td>-</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>LimitedStops</td>
<td>N</td>
<td>-</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>HailAndRide</td>
<td>Y</td>
<td>N</td>
<td>-</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>FlexibleService</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ExcursionOrTour</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>OtherService</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Purpose</td>
<td>SchoolOrWorks</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>RuralService</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6-24 – Allowed ServiceClassification Combinations

6.6.6.6 Service / ServiceOperational Group

The ServiceOperationalGroup (Figure 6-56) group holds ancillary descriptive elements describing operational data associated with the Service.

- **TicketMachineServiceCode**: Unique Identifier associated with the Service for use in ticketing machine systems. May be overridden on Individual Journey Patterns & Vehicle Journey instances.
- **VehicleType**: The type of vehicle normally used on the service – can be used to specify if the Service is normally considered to be wheelchair accessible. More specific details may be
provided by the **VehicleType** / **Wheelchair** **VehicleEquipment**. (+TXC v2.4) See later below.

- **AssistanceService**: The assistance normally available on the service – can be overridden on individual journeys (+TXC v2.5). See later below.

- **ServiceFacilitySet**: The facilities normally available on the service – can be overridden on individual journeys (+TXC v2.5). See later below.

---

The **AssistanceService** element (Figure 6-64) describes the available assistance for wheelchair users to access a vehicle. (+TXC V2.5)

- **Name**: Description of service. (+TXC v2.5)

- **Availability**: Days or times when service is available. (+TXC v2.5.) If not specified assume always available.
  - **DayType**: Day type and Timeband when service is available.

- **AssistanceServiceType**: Assistance services for wheelchair users. See Table 6-25. (+TXC v2.5.)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No Assistance service is available</td>
</tr>
<tr>
<td>boardingAssistance</td>
<td>Assistance is available to board the vehicle.</td>
</tr>
<tr>
<td>wheelchairAssistance</td>
<td>Assistance is available for wheelchair users</td>
</tr>
<tr>
<td>conductor</td>
<td>A conductor is available</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known if available.</td>
</tr>
</tbody>
</table>

**Table 6-25 – Allowed Values for AssistanceServiceType (+TXC v2.5)**

- **AssistanceAvailability**: Availability of services for wheelchair users. Table 6-8 earlier shows the allowed values for assistance availability. (+TXC v2.5.)

- **Staffing**: shows the normal staffing arrangements for the stop. See **Table 6-26** for allowed values

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullTime</td>
<td>Staffed full time during opening hours</td>
</tr>
<tr>
<td>partTime</td>
<td>Only staffed at certain times</td>
</tr>
<tr>
<td>unmanned</td>
<td>Not normally staffed</td>
</tr>
<tr>
<td>unknown</td>
<td>Staffing unknown</td>
</tr>
</tbody>
</table>

**Table 6-26 – Allowed Values for Staffing (+TXC v2.5)**

- **AccessibilityTools**: Devices available for use by passengers. One or more - See Table 6-27 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wheelchair</td>
<td>Wheelchairs are available for use.</td>
</tr>
</tbody>
</table>
walkingStick  |  Walking sticks are available for use.
audioNavigator  |  Audio Navigators are available for use.
visualNavigator  |  Visual Navigator aids are available for use, e.g. floor strips.
passengerCart  |  Passenger carts are available for use.
other  |  Other equipment is available for use.

Table 6-27 – Allowed Values for AccessibilityTools (+TXC v2.5)

- **Languages**: Languages spoken for assistance. One or more languages, as specified by XML Language value. (e.g. en, cy, de, fr, etc). (+TXC v2.5.)
- **AccessibilityTrainedStaff**: Staff is trained to assist wheelchair users. (+TXC v2.5.)
- **EmergencyServices**: List of available emergency service support. One or more - See Table 6-28 for allowed values. (+TXC v2.5.)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>police</td>
<td>Police are available at location or on board.</td>
</tr>
<tr>
<td>firstAid</td>
<td>First aid is available at location or on-board.</td>
</tr>
<tr>
<td>sosPoint</td>
<td>SOS help Points are available at location or on board.</td>
</tr>
<tr>
<td>ccTv</td>
<td>CCTV covers location or on board.</td>
</tr>
<tr>
<td>other</td>
<td>Other services available at location or on board.</td>
</tr>
</tbody>
</table>

Table 6-28 – Allowed Values for EmergencyServices (+TXC v2.5)
The **ServiceFacilitySet** element (Figure 6-65) describes the available on-board services for a vehicle as simple lists of named facilities. Certain services with complex properties can be additionally described by an equipment element, for example **SanitaryFacilityEquipment** allows detailed properties of **SanitaryEquipment** to be specified. (+TXC V2.5).

- **Name**: Description of service. (+TXC v2.5.)
- **CommonFacilityGroup** describes services that may also apply to a Site:
  - **CateringFacilityList**: Catering users. See **Table 6-29**. (+TXC v2.5.)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>No Assistance service is available</td>
</tr>
<tr>
<td>boardingAssistance</td>
<td>Assistance is available to board the vehicle.</td>
</tr>
<tr>
<td>wheelchairAssistance</td>
<td>Assistance is available for wheelchair users</td>
</tr>
<tr>
<td>conductor</td>
<td>A conductor is available</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known if available</td>
</tr>
</tbody>
</table>
Table 6-29 – Allowed Values for CateringFacilityList

- **FareClasses**: (+TXC v2.5) specifies the classes of fare available. See Table 6-30 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>firstClass</td>
<td>First class seats</td>
</tr>
<tr>
<td>secondClass</td>
<td>Second or standard class seats</td>
</tr>
<tr>
<td>thirdClass</td>
<td>Third class seat</td>
</tr>
<tr>
<td>businessClass</td>
<td>Business class</td>
</tr>
<tr>
<td>economyClass</td>
<td>Economy Class</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known if available.</td>
</tr>
</tbody>
</table>

Table 6-30 – Allowed values for ServiceFacilitySet / FareClasses (+TXC v2.4)

- **NuisanceFacilityList**: (+TXC v2.5) The type of accommodation available. See Table 6-31 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smoking</td>
<td>Smoking zone.</td>
</tr>
<tr>
<td>noSmoking</td>
<td>No smoking. zone.</td>
</tr>
<tr>
<td>mobilePhoneUseZone</td>
<td>Mobile phone zone.</td>
</tr>
<tr>
<td>mobilePhoneFreeZone</td>
<td>Quiet area.</td>
</tr>
<tr>
<td>familyArea</td>
<td>Family area.</td>
</tr>
<tr>
<td>childfreeArea</td>
<td>Child free area.</td>
</tr>
<tr>
<td>other</td>
<td>Other area.</td>
</tr>
<tr>
<td>smoking</td>
<td>Smoking zone.</td>
</tr>
<tr>
<td>noSmoking</td>
<td>No Smoking. zone.</td>
</tr>
<tr>
<td>mobilePhoneUseZone</td>
<td>Mobile phone zone.</td>
</tr>
<tr>
<td>mobilePhoneFreeZone</td>
<td>Quiet area.</td>
</tr>
<tr>
<td>familyArea</td>
<td>Family area.</td>
</tr>
</tbody>
</table>

Table 6-31 – Allowed values for ServiceFacilitySet / NuisanceFacility (+TXC v2.5)

- **PassengerCommsFacilityList**: (+TXC v2.5) The types of on-board passenger communications and entertainment facilities. See Table 6-32 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>freeWifi</td>
<td>Free wifi.</td>
</tr>
<tr>
<td>publicWifi</td>
<td>Public wifi, may be charged.</td>
</tr>
<tr>
<td>telephone</td>
<td>Telephone.</td>
</tr>
<tr>
<td>internet</td>
<td>Internet access.</td>
</tr>
<tr>
<td>mobileCoverage</td>
<td>Mobile phone coverage.</td>
</tr>
<tr>
<td>videoEntertainment</td>
<td>Video entertainment.</td>
</tr>
<tr>
<td>audioEntertainment</td>
<td>Audio entertainment.</td>
</tr>
<tr>
<td>powerSupplySockets</td>
<td>Power supply sockets.</td>
</tr>
</tbody>
</table>

Table 6-32 – Allowed values for ServiceFacilitySet / PassengerCommsFacilityList (+TXC v2.5)

- **PassengerInformationFacilityList**: (+TXC v2.5) The type of passenger information facilities available on-board. See Table 6-33 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioInformation</td>
<td>Audio information.</td>
</tr>
<tr>
<td>audioForHearingImpaired</td>
<td>Special Audio information for those who are hearing impaired.</td>
</tr>
<tr>
<td>visualDisplays</td>
<td>Visual displays of information.</td>
</tr>
<tr>
<td>displaysForVisuallyImpaired</td>
<td>Specially enhanced visual information for those who are visually impaired.</td>
</tr>
<tr>
<td>largePrintTimetables</td>
<td>Large print timetables.</td>
</tr>
<tr>
<td>other</td>
<td>Other INFO FACILITY.</td>
</tr>
</tbody>
</table>

Table 6-33 – Allowed values for ServiceFacilitySet / PassengerInformationFacility (+TXC v2.5)

- **SafetyFacilityList**: (+TXC v2.5) Some on-board capabilities that enhance to passenger safety. See Table 6-34 for allowed values.
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mobileCoverage</td>
<td>There is mobile phone coverage.</td>
</tr>
<tr>
<td>cctV</td>
<td>There is close circuit television camera surveillance.</td>
</tr>
<tr>
<td>sosPoint</td>
<td>There is an SOS point.</td>
</tr>
<tr>
<td>staffed</td>
<td>There are staff.</td>
</tr>
</tbody>
</table>

Table 6-34 – Allowed values for ServiceFacilitySet / SafetyFacility (+TXC v2.5)

- **SanitaryFacilityList**: (+TXC v2.5) The types of lavatory and washing facilities available on service. See Table 6-35 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>toilet</td>
<td>Lavatories.</td>
</tr>
<tr>
<td>wheelchairAccessToilet</td>
<td>Toilet for wheelchair access.</td>
</tr>
<tr>
<td>shower</td>
<td>Shower.</td>
</tr>
<tr>
<td>wheelchairBabyChange</td>
<td>Baby change facility for wheelchair users.</td>
</tr>
<tr>
<td>babyChange</td>
<td>Baby change facility.</td>
</tr>
<tr>
<td>washingAndChangeFacilities</td>
<td>Wash and Change facilities.</td>
</tr>
<tr>
<td>other</td>
<td>Other sanitary facility.</td>
</tr>
</tbody>
</table>

Table 6-35 – Allowed values for ServiceFacilitySet / SanitaryFacility (+TXC v2.5)

- **ServiceFacilityGroup** describes services specific to a Service:

- **AccommodationFacilityList**: (+TXC v2.5) The type of accommodation available. See Table 6-36 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seating</td>
<td>Seating.</td>
</tr>
<tr>
<td>sleeper</td>
<td>Sleeper accommodation.</td>
</tr>
<tr>
<td>double Sleeper</td>
<td>Double Sleeper accommodation.</td>
</tr>
<tr>
<td>single Sleeper</td>
<td>Single Sleeper accommodation.</td>
</tr>
<tr>
<td>special Sleeper</td>
<td>Special Sleeper accommodation.</td>
</tr>
<tr>
<td>couchette</td>
<td>Couchette.</td>
</tr>
<tr>
<td>reclining Seat</td>
<td>Reclining seat.</td>
</tr>
<tr>
<td>babyCompartment</td>
<td>Baby Compartment.</td>
</tr>
<tr>
<td>familyCompartment</td>
<td>Family Compartment.</td>
</tr>
<tr>
<td>panoramaCoach</td>
<td>Panorama coach.</td>
</tr>
<tr>
<td>pullmanCoach</td>
<td>Pullman coach.</td>
</tr>
<tr>
<td>standing</td>
<td>Standing accommodation.</td>
</tr>
</tbody>
</table>

Table 6-36 – Allowed values for ServiceFacilitySet / AccommodationFacility (+TXC v2.5)

- **LuggageCarriageFacilityList**: (+TXC v2.5) describes the type of l available. See Table 6-37 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noBaggageStorage</td>
<td>No baggage storage.</td>
</tr>
<tr>
<td>luggageRacks</td>
<td>Luggage racks.</td>
</tr>
<tr>
<td>extraLargeLuggageRacks</td>
<td>Extra-large luggage storage.</td>
</tr>
<tr>
<td>baggageVan</td>
<td>Baggage van.</td>
</tr>
<tr>
<td>cyclesAllowed</td>
<td>Cycles allowed.</td>
</tr>
<tr>
<td>cyclesAllowedInVan</td>
<td>Cycles allowed in van.</td>
</tr>
<tr>
<td>cyclesAllowedInCarriage</td>
<td>Cycles allowed in carriage.</td>
</tr>
<tr>
<td>cyclesAllowedWithReservation</td>
<td>Cycles allowed with reservation.</td>
</tr>
<tr>
<td>noCycles</td>
<td>No cycles allowed.</td>
</tr>
<tr>
<td>noBaggageStorage</td>
<td>No baggage storage.</td>
</tr>
<tr>
<td>luggageRacks</td>
<td>Luggage racks.</td>
</tr>
<tr>
<td>extraLargeLuggageRacks</td>
<td>Extra-large luggage storage.</td>
</tr>
</tbody>
</table>

Table 6-37 – Allowed values for ServiceFacilitySet / LuggageCarriage (+TXC v2.5)

- **ReservationFacilityList**: (+TXC v2.5) Date that Service starts.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seatReservationsCompulsory</td>
<td>Seat Reservations Compulsory</td>
</tr>
<tr>
<td>cycleReservationsCompulsory</td>
<td>Bicycle Reservations Compulsory</td>
</tr>
<tr>
<td>seatReservationsRecommended</td>
<td>Seat Reservations Recommended</td>
</tr>
<tr>
<td>seatReservationsPossible</td>
<td>Seat Reservations Possible</td>
</tr>
<tr>
<td>wheelchairOnlyReservations</td>
<td>Reservations are required for Wheelchair but not necessarily for others</td>
</tr>
</tbody>
</table>
### Table 6-38 – Allowed values for ServiceFacilitySet / Reservations (+TXC v2.4)

<table>
<thead>
<tr>
<th>ServiceFacilitySet_VersionStructure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreatingFacilityList</td>
<td>Description of facility set.</td>
</tr>
<tr>
<td>FareClasses</td>
<td>List of fare classes.</td>
</tr>
<tr>
<td>NuisanceFacilityList</td>
<td>List of nuisance facilities.</td>
</tr>
<tr>
<td>PassengerCommsFacilityList</td>
<td>List of passenger communications facilities.</td>
</tr>
<tr>
<td>PassengerInformationFacilityList</td>
<td>List of passenger information facilities.</td>
</tr>
<tr>
<td>SafetyFacilityList</td>
<td>List of safety facilities.</td>
</tr>
<tr>
<td>SanitaryFacilityList</td>
<td>List of sanitary facilities.</td>
</tr>
<tr>
<td>AccommodationFacilityList</td>
<td>List of accommodation facilities.</td>
</tr>
<tr>
<td>LuggageCarriageFacilityList</td>
<td>List of luggage carriage facilities.</td>
</tr>
<tr>
<td>ReservationFacilityList</td>
<td>List of reservation facilities.</td>
</tr>
</tbody>
</table>

### Figure 6-65 – ServiceFacilitySet Element (+TXC v2.5)

6.6.6.9 Service / AssociatedOperators Element

The **AssociatedOperators** (Figure 6-66) element records details about any operators associated with the service other than the registered operator. The **AssociatedOperator** comprises:

- **OperatorRef**: Reference to an **Operator** or **LicensedOperator** definition. See above.
- **Role**: Description of the role of the associated operator.

### Figure 6-66 – Service / AssociatedOperators Element

6.6.6.10 Service / StopRequirements Element

The **StopRequirements** element (Figure 6-67) specifies whether a service does or does not require any new stops, and provides a list of these stops. If not, specifies, default is to assume **NoNewStopsRequired**. **StopRequirements** does not have to be specified on a Cancellation.

- **NoNewStopsRequired**: No new stops are needed.
- **NewStops**: New stops are needed.
  - **StopPointRef**: Reference to identify the new stop.
  - **Note**: Optional explanatory note accompanying definition.
Note that an explicit list of the new stops associated with the service is useful to the EBSR registration authorities. That a stop is relevance to a particular service cannot be inferred merely from the fact that a new stop has been added to NaPTAN. The NewStops tag should list the identifiers of the new stops. Details of the stops may already have been added to NaPTAN or may be declared locally.

### 6.6.6.11 Service / ServiceAvailability Element

The **ServiceAvailability** element (Figure 6-68) specifies the time of day a service runs (TIME DEMAND TYPE) as a broad classification. One of the following:

- **TwentyFourHours**: Service runs all day and all night continuously.
- **Daytime**: Service runs in daytime.
- **Peak**: Service runs in peak hours only.
- **OffPeak**: Service runs in off-peak hours only.
- **Night**: Service is a night service.

### 6.6.6.12 Service / ToBeMarketedWith Element

The **ToBeMarketedWith** element (Figure 6-69) records the Services that are normally marketed with the bus service. It contains one or more **RelatedService** instances, each of which is an **AnnotatedServiceRefStructure**.

- **ServiceRef**: Reference to another **Service** definition provided elsewhere in the document.
- **Description**: Text description of the service &/or its identifier if not defined by a service reference.
6.6.7 Service / JourneyGroupings (TXC v2.4)

The *JourneyGroupings* element (Figure 6-70) allows Journey Groupings for the service to be declared. Each of these may have a description and be associated the vehicle journeys of the service. There are six built in *JourneyGroupings* with which journeys will be associated by default. Other Custom Groupings may be added as overrides.

Each *JourneyGroupings* has three sets of JourneyGrouping instances:

- **InboundJourneyGroupings**: Built-in Inbound *JourneyGroupings*. See below.
- **CustomJourneyGroupings**: Built-in Custom *JourneyGroupings*. See below.

### Figure 6-70 – Service / JourneyGroupings Element

#### 6.6.7.1 Service / Built In Journey Grouping Element (TXC v2.4)

The *BuiltInJourneyGrouping* Element (Figure 6-71) specifies the properties of an individual built in *JourneyGrouping*. Each *JourneyGrouping* has:

- **Description**: Description of *JourneyGrouping* (to be shown on matrix bed).
- **Vias**: List of Via Names of *JourneyGrouping* (to be shown on matrix bed).
  - **Via**: Name of a place in *Vias* list.
- **StopPointRef**: Designated stop to use to sort columns of *JourneyGrouping* by Journey times. (Note yet supported by Publisher – will be ignored if there are Sequence numbers on Journeys corresponding to matrix columns).
- **Notes**: Notes associated with *JourneyGrouping*.
- **Contents**: List of Via Names of *JourneyGrouping* (to be shown on matrix bed).
  - **ByOperationalProfile**: Include any vehicle journeys that have the same day type and direction and are not in a *CustomJourneyGrouping*.
  - **None**: Select journeys as for *ByOperationalProfile*, but then suppress this grouping from the published timetable.
AbstractJourneyGroupingStructure (extension)

BuiltinJourneyGroupingStructure

Type for Built in Journey Grouping on a specific Day Type/Direction.

GenerateTitle

Type: xsd:boolean

Whether a title should be generated from the OperatingProfile and direction, for example, 'Outbound, Monday To Friday'.

Description

Type: NaturalLanguageStringStructure

Text description of route to use for journey grouping. Will be appended to any generated title.

ViasStructure

A list of place names to use to create journey grouping description. Will be published as A-B-C-D etc.

Vias

Type: ViasStructure

Name of intermediate point to present as Via point from this stop.

Via

Type: NaturalLanguageStringStructure

Note

Type: NoteStructure

Notes associated with Journey grouping.

ByOperationalProfile

Type: EmptyType

Suppress this grouping.

None

Type: EmptyType

No Via should be shown - suppress any from parent. (TXC 2.4)

StopPointRef

Reference to a Stop Point whose departure times can be used for horizontal sorting of columns within matrix. (Not yet supported by Publisher in TXC 2.4)

Note

Type: NoteStructure

Notes associated with Journey grouping.

Contents

Criteria for including

ByOperationalProfile

Type: EmptyType

Suppress this grouping.

None

Type: EmptyType

No Via should be shown - suppress any from parent. (TXC 2.4)

Figure 6-71 – Service / Built in Journey Grouping Element

6.6.7.2 Service / CustomJourneyGrouping Element

The CustomJourneyGrouping element (Figure 6-58) specifies the properties of an individual built in JourneyGrouping. A Custom Journey Group element shares common Properties with a Built-in JourneyGrouping.

In addition, you may specify:

- **Direction**: Direction of Journeys.
- **PrivateKey**: Optional preferred colour to use when showing line in graphic media.
- **VehicleJourneyRefs**: List of references to VehicleJourney instances that are in the CustomJourneyGrouping.
  - **VehicleJourneyRef**: Reference to an individual VehicleJourney.
CustomJourneyGroupingStructure

CustomJourneyGrouping

( extension)

GenerateTitle

Whether a title should be generated from the OperatingProfile and direction, for example, 'Outbound, Monday to Friday'

Description

Text description of route to use for journey grouping, will be appended to any generated title

Vias

A list of place names to use to create journey group description. Will be published as A-B-C-D etc.

StopPointRef

Reference to a Stop Point whose departure times can be used for horizontal sorting of columns within matrix. (Not yet supported by Publisher in v TXC 2.4)

Notes

Notes associated with journey grouping.

CustomJourneyGrouping

Elements defining a custom Journey Grouping

Direction

PrivateCode

VehicleJourneyRefs

VehicleJourneyRef

Extensions

ExtensionsAnyStructure

Figure 6-72 – Service / CustomJourneyGrouping Element
6.7 StandardService, JourneyPattern, VehicleJourney

6.7.1 StandardService Element

The **StandardService** element (Figure 6-73) describes the fixed-route component of a **Service**. It comprises:

- **ServicePartGroup** – elements common to standard and flexible journey parts.
  - **Origin**: Public name of the place where the service starts.
  - **Destination**: Public name of the place where the service ends.
  - **Vias**: Public name(s) of the places that the service route goes past: One or more **Via** elements. See below.
  - **UseAllStopPoints**: Whether the service uses all the stops along its **Route**.
  - **ReturnVias**: Public names of main points on opposite direction. If absent assumed to be the Vias in inverse order. (TXC v2.4)

**StandardServiceGroup** – elements specific to **StandardService** parts.

- **JourneyPattern**: One or more **JourneyPattern** elements representing the working of the service. See below.
- **Cancellation**: If registration is a cancellation, **JourneyPattern** may be omitted.

![StandardService Structure](image)

**Figure 6-73 – StandardService Element**
6.7.2 StandardService / Subelements

6.7.2.1 StandardService / Vias Element

The **Vias** element (Figure 6-58) allows an indication of the routing of the service as a sequence of text names. It is used to generate the default service description for Matrix beds when publishing the service (reversed for the inbound/outbound directions). It may be overridden (TXC 2.4) using a **JourneyGrouping**.

Each **Vias** has:
- One or more **Via** names.
- Or **None**: No **Via** Names should be shown. (TXC v2.4)

![Figure 6-74 – Service / Vias Element](image-url)
6.7.3 JourneyPatterns

A JourneyPattern describes a possible bus route of a StandardService as a sequence of timing links between stops that a vehicle will traverse in a particular order, representing the pattern of working for vehicles of the service.

- Each JourneyPattern belongs to a StandardService.
- The individual steps of the journey are modelled as JourneyPatternTimingLink elements; each link has information about the distance to travel, between two stops, and the run time needed. This is to enable the computation of accurate journey timings to each stop. Activity at stop and other information about stop usage is described for each end of the link using JourneyPatternStopUsage elements.
- The links are grouped into JourneyPatternSection elements, representing reusable link sequences. Sections are declared within a TransXChange top-level container element, JourneyPatternSections, and so may be reused in different JourneyPattern instances.
- The order of JourneyPatternTimingLinks in each JourneyPatternSection, and the overall order of the JourneyPatternSection instances must both follow the order in which they are traversed.
- The timing links of a JourneyPattern should correspond to the RouteLink instances of any associated Route, that is be an exact projection on a link-by-link basis of either all the links of route in sequence, or a contiguous subset of the route links in sequence
- In a given JourneyPattern, the route links of an individual RouteSection should all be referenced by timing links in a single JourneyPatternSection, i.e. not be divided between different JourneyPatternSection instances. A JourneyPatternSection may however project onto multiple RouteSection instances.
- A JourneyPattern may be used in more than one VehicleJourney on a route. It should be noted that a VehicleJourney following a JourneyPattern may not necessarily stop at all stops identified within the JourneyPattern, thus the JourneyPattern provides the ‘super set’ of stops of a route, of which all or some may be served by the dependent VehicleJourney instances. Individual VehicleJourney instances may subset the full JourneyPattern stop list either by passing an individual stop, or by short working at either end. They must still follow the route and stop sequence for the part of the journey pattern that they work.

6.7.4 JourneyPattern Element

A JourneyPattern (Figure 6-75) describes the stopping pattern of a standard i.e. fixed route service. A JourneyPattern is identified by a unique id attribute, and comprises a number of elements falling into two groups:

1. **CommonJourneyGroup**: Shared elements common to journey patterns and vehicle journeys.
2. **JourneyPatternGroup**: Elements specific to journey patterns.

![Figure 6-75 – JourneyPattern Element](image-url)
6.7.4.1 JourneyPattern / CommonJourneyGroup

The **CommonJourneyGroup** (Figure 6-76) holds identity and operational information that is common to both a **JourneyPattern** and a **VehicleJourney**: the **JourneyPattern** instances provide default values to use on dependent **VehicleJourney** instances if no specific override is provided on the **VehicleJourney**.

- **JourneyIdentificationGroup**: Elements for identifying the journey – see below.
- **OperatingProfile**: Specifies operational days and times associated with the JourneyPattern. If not specified inherited from **Service**.
- **Operational**: Specifies additional operational information associated with the journey. See below. Normally this is not required since it is the same as for the service. Includes **TicketMachine** and **Block** elements. See below.
- **TimeDemand**: Classification of the route as to when peak demand occurs. See Table 6-39.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>earlyMorning</td>
<td>Early Morning.</td>
</tr>
<tr>
<td>offPeak</td>
<td>Off Peak.</td>
</tr>
<tr>
<td>peakMorning</td>
<td>Peak Morning.</td>
</tr>
<tr>
<td>peakAfternoon</td>
<td>Peak Afternoon.</td>
</tr>
<tr>
<td>evening</td>
<td>Evening.</td>
</tr>
<tr>
<td>lateEvening</td>
<td>Late Evening.</td>
</tr>
<tr>
<td>saturdayMorning</td>
<td>Saturday Morning.</td>
</tr>
<tr>
<td>saturdayDaytime</td>
<td>Saturday Daytime.</td>
</tr>
<tr>
<td>saturdayEvening</td>
<td>Saturday Evening.</td>
</tr>
<tr>
<td>sunday</td>
<td>Sunday.</td>
</tr>
<tr>
<td>bankHoliday</td>
<td>Bank Holiday.</td>
</tr>
</tbody>
</table>

Table 6-39 – Allowed Values for TimeDemand

- **CommercialBasis**: on which the service is offered. May be overridden for specific journey patterns, vehicle Journeys and timing links. See Table 6-40 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contracted</td>
<td>Service on Link is contracted</td>
</tr>
<tr>
<td>notContracted</td>
<td>Service on Link is not contracted</td>
</tr>
<tr>
<td>partContracted</td>
<td>Service on Link is part contracted</td>
</tr>
<tr>
<td>inherit</td>
<td>Basis is same as that of parent service or journey pattern</td>
</tr>
<tr>
<td>unknown</td>
<td>Basis is unknown</td>
</tr>
</tbody>
</table>

Table 6-40 – Allowed Values for Service / CommercialBasis.

- **LayoverPoint**: Points at which the service lays over. See below.
- **GarageRef**: A garage from which the **Service** operates.
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.

The JourneyIdentificationGroup holds information for identifying a journey.

- **RouteRef**: The Route which the JourneyPattern follows. See Route above.
- **JourneyPatternSectionRefs**: An ordered collection of references to JourneyPatternSections (as JourneyPatternSectionRef instances), that contain the journey pattern timing links making up the JourneyPattern. See JourneyPatternSection later.
Journey Identification Properties common to both journey patterns and vehicle journeys.

Figure 6-77 – JourneyIdentificationGroup

6.7.5 JourneyPattern Subelements

6.7.5.1 JourneyPattern / StandardJourneyPatternGroup

The StandardJourneyPatternGroup (Figure 6-77) holds information to identify a journey:

- **PrivateCode:** A unique private code that can be used to identify the JourneyPattern.
- **DestinationDisplay:** Journey destination, as displayed on vehicle. If omitted, the Destination of the Service is used.
- **OperatorRef:** The operator for the journey. Normally this is not required since it is the same as for the Service.
- **Direction:** The default Direction of the JourneyPattern. Default is ‘inherit’. See Table 6-41 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherit</td>
<td>Use value from Service.</td>
</tr>
<tr>
<td>inbound</td>
<td>Inbound Direction.</td>
</tr>
<tr>
<td>outbound</td>
<td>Outbound Direction.</td>
</tr>
<tr>
<td>clockwise</td>
<td>Clockwise Direction.</td>
</tr>
<tr>
<td>antiClockwise</td>
<td>Anti-Clockwise Direction.</td>
</tr>
</tbody>
</table>

Table 6-41 – Allowed Values for JourneyPattern / Direction

Properties specific to Standard Journey Patterns.

Figure 6-78 – JourneyPattern / StandardJourneyPatternGroup

6.7.5.2 CommonJourneyGroup / Operational / OperationalVariant Element

The OperationalVariant element (Error! Reference source not found.) specifies operational information variants for a particular day associated with the JourneyPattern:

- Day on which the variant applies specified either as:
  - **OperationalProfile:** In line description of day properties
**OperationalGroup**: Data elements that relate to operational aspects of the journey: as above.

- **DayTypeRef**: Reference to a `DayType`. See Calendar/DayType later below.

**OperationalVariantStructure**

---

**Figure 6-79 – JourneyPattern / Operational / OperationalVariant Element**

6.7.5.3 CommonJourneyGroup / Block Element

The **Block** element (Figure 6-80) specifies information about the block (running board) of a journey. A block enables **VehicleJourney** instances to be assigned to a logical group of journeys that will be carried out by the same vehicle.

- **Description**: Text describing the block.
- **BlockNumber**: The number of the block associated with the journey. **VehicleJourney** instances with the same **BlockNumber** will be carried out by the same vehicle.
- **Note**: Explanatory text to explaining any further operational particulars about the block.

**Figure 6-80 – Operational / Block Element**

6.7.5.4 CommonJourneyGroup / VehicleType Element

The **VehicleType** element (Figure 6-81) describes a type of vehicle running a service.

- **VehicleTypeCode**: Arbitrary code that classifies the vehicle.
- **Description**: Free text description of vehicle type.
- **VehicleEquipment**: Different types of equipment available on the Vehicle.
  - **PassengerInfoEquipment**: Information relating to passenger Information facilities on the vehicle. (+TXC V2.4)
  - **AccessEquipment**: Information relating to data access on the vehicle. (+TXC V2.4)
  - **WheelchairEquipment**: Information relating to wheelchair access on the vehicle. (+TXC V2.4)
Detailed properties of on vehicle services are described with different types of Equipment. All types of equipment share common Equipment element properties (Figure 6-83) (+TXC V2.5)

- **Name**: Name of Equipment (+TXC v2.5).
- **Availability**: Condition specifying when equipment is available (+TXC v2.5).
  - **DaysOfWeek**: Which days of week equipment is available
  - **Timebands**: When equipment is available.

The PassengerInfoEquipment element (Figure 6-83) describes a type of vehicle equipment relating to passenger information service. (+TXC V2.4)

- **PassengerInfo**: Arbitrary code that classifies the vehicle. See Table 6-40 for allowed values.
### Table 6-42 – Allowed Values for PassengerInfoEquipment / PassengerInfo.

- **AccessibilityInfo**: Arbitrary code that classifies the vehicle. See Table 6-40 for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>audioInformation</td>
<td>Vehicle has audioInformation</td>
</tr>
<tr>
<td>audioForHearingImpaired</td>
<td>Vehicle has audioForHearingImpaired</td>
</tr>
<tr>
<td>visualDisplays</td>
<td>Vehicle has visualDisplays</td>
</tr>
<tr>
<td>displaysForVisuallyImpaired</td>
<td>Vehicle has displaysForVisuallyImpaired</td>
</tr>
<tr>
<td>tactilePlatformEdges</td>
<td>Vehicle has tactilePlatformEdges</td>
</tr>
<tr>
<td>tactileGuidingStrips</td>
<td>Vehicle has tactileGuidingStrips</td>
</tr>
<tr>
<td>largePrintTimetables</td>
<td>Vehicle has largePrintTimetables</td>
</tr>
<tr>
<td>other</td>
<td>other</td>
</tr>
</tbody>
</table>

## Figure 6-83 – Operational / Equipment Element

The **AccessVehicleEquipment** element (Figure 6-84) describes a type of vehicle equipment relating to physical access to the vehicle relevant to accessibility planning. (+TXC V2.4)

- **LowFloor**: Whether Vehicle has low floor access.
- **Hoist**: Whether Vehicle has hoist for wheelchair access. (+TXC V2.5)
- **Ramp**: Whether Vehicle has deployable ramp.
- **RampBearingCapacity**: Whether Vehicle has deployable ramp.
- **NumberOfSteps**: Number of steps to board vehicle.
- **BoardingHeight**: Rise from ground needed for 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.
- **SuitableFor**: Mobility need for which access is suitable. See Table 6-43 below. (+TXC v2.5)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wheelchair</td>
<td>Wheelchair</td>
</tr>
<tr>
<td>assistedWheelchair</td>
<td>Wheelchair pushed by companion</td>
</tr>
<tr>
<td>motorisedWheelchair</td>
<td>Motorised Wheelchair</td>
</tr>
<tr>
<td>mobilityScooter</td>
<td>Small mobility Scooter: A Class 2 scooter under the CPT classification with 3 or 4 wheels, not more than 600mm wide and 1000 mm long and with a turning radius not exceeding 1200mm. Normally weigh about 65 kg.</td>
</tr>
<tr>
<td>roadMobilityScooter</td>
<td>Large Mobility Scooter: A Class 3 scooter under the CPT classification. Class 3 scooters are bigger and have light for road use. They are not normally allowed on buses.</td>
</tr>
<tr>
<td>walkingFrame</td>
<td>Walking Frame</td>
</tr>
<tr>
<td>restrictedMobility</td>
<td>Restricted Mobility</td>
</tr>
<tr>
<td>normal</td>
<td>Normal mobility</td>
</tr>
</tbody>
</table>

Table 6-43 – Allowed Values for MobilityNeed

- **AssistanceNeeded**: Nature of assistance needed to board See (Table 6-44). (+TXC v2.5)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>levelAccess</td>
<td>Level access – passenger can propel themselves</td>
</tr>
<tr>
<td>rampRequired</td>
<td>Assistance with ramp needed.</td>
</tr>
<tr>
<td>hoistRequired</td>
<td>Assistance with hoist needed.</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known.</td>
</tr>
</tbody>
</table>

Table 6-44 – Allowed Values for AssistanceNeeded

- **AssistedBoardingLocation**: Whether boarding has to be done at a specific position on the platform. See Table 6-45). (+TXC v2.5)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boardAtAnyDoor</td>
<td>Boarding can be at any location</td>
</tr>
<tr>
<td>boardOnlyAtSpecifiedPositions</td>
<td>Boarding must be at specific positions on platform</td>
</tr>
<tr>
<td>unknown</td>
<td>Not known.</td>
</tr>
</tbody>
</table>

Table 6-45 – Allowed Values for AssistedBoardingLocation

- **GuideDogsAllowed**: Whether guide dogs are allowed. (+TXC v2.5)
### Access Vehicle Equipment Structure

#### Attributes
- **Name** (`xsd:string`): National language string structure.
- **Availability** (`xsd:string`): When service is available. (+TXC 2.5)

#### Extensions to schema
- **Extensions** (`xsd:anySimpleType`): Wrapper tag used to avoid problems with handling of optional 'any' by some validators.

#### Elements
- **LowFloor** (`xsd:boolean`): Whether vehicle is low floor.
- **Hoist** (`xsd:boolean`): Whether vehicle has a hoist or lift for wheelchairs.
- **Ramp** (`xsd:boolean`): Whether there is a ramp to access vehicle.
- **RampBearingCapacity** (`xsd:positiveInteger`): Maximum weight that ramp can bear.
- **NumberOfSteps** (`xsd:nonNegativeInteger`): Number of steps to board or alight from vehicle.
- **BoardingHeight** (`xsd:decimal`): Maximum step height to board.
- **GapToPlatform** (`xsd:decimal`): Normal gap between vehicle and platform.
- **WidthOfAccessArea** (`xsd:decimal`): Width of access area.
- **HeightOfAccessArea** (`xsd:decimal`): Height of access area.
- **AutomaticDoors** (`xsd:boolean`): Whether there are Automatic Doors.
- **SuitableFor** (`xsd:string`): Mobility needs for which access is suitable.
- **AssistanceNeeded** (`xsd:positiveInteger`): Nature of assistance needed to board - level access allows self-boarding.
- **AssistedBoardingLocation** (`xsd:string`): Whether special position on platform is needed for boarding.

#### Figure 6-84 – VehicleType/AccessVehicleEquipment Element (+TXC V2.4)

6.7.5.8 VehicleType / WheelchairEquipment Element (+TXC V2.4)

The **WheelchairEquipment** element (Figure 6-85) describes a type of on board vehicle equipment for wheelchair access on the vehicle. (+TXC V2.4)

- **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**: Wheelchair turning circle on board.
- **SuitableFor**: Mobility need for which access is suitable, for example motorised wheelchair, mobility scooter, etc. See Table 6-43 earlier. (+TXC v2.5.)
- **BookingRequired**: Whether booking is required for wheelchair use. (+TXC v2.5.)
- **BookingNumber**: Phone number to call for wheelchair booking (+TXC v2.5.)

**Figure 6-85 – VehicleType/ WheelchairVehicleEquipment Element (+TXC v2.4)**

The **TicketMachine** element (Figure 6-86) specifies information for associating a journey with the settings of a ticket machine.

- **TicketMachineServiceCode**: Unique Identifier associated with service for use in ticketing machine systems. If not specified, defaults to any value specified at the Service Level.
- **JourneyCode**: The identifier used by the ticket machine system to refer to the journey.
- **Direction**: The direction used by the ticket machine system to refer to the journey.
The LayoverPoint element (Figure 6-87) describes a layover point used in a journey pattern. It is identified by an id attribute, and comprises:

- **Duration**: Time of wait at layover point. Uses standard duration type.
- **Name**: Free text description of layover point.
- **Location**: Location of layover point.
- **Minimum Duration**: Minimum time for a layover at a point (+v2.4)

A JourneyPatternSection (Figure 6-88) declares and groups an ordered collection of JourneyPatternTimingLink elements. Each JourneyPatternSection can be identified by a unique id attribute.
6.7.7 JourneyPatternTimingLink Element

A JourneyPatternTimingLink (Figure 6-89) describes a timed link connecting two stops of a JourneyPattern of a StandardService. It allows a precise default timing to be set for traversing a link. Each JourneyPatternTimingLink can be identified by a unique id attribute, and comprises a number of elements falling into two groups:

1. **CommonTimingLinkGroup**: Shared elements common to journey pattern timing links and to vehicle journey timing links.
2. **JourneyPatternTimingLinkGroup**: Elements specific to journey pattern timing links.

### 6.7.7.1 JourneyPatternTimingLink / CommonTimingLinkGroup

The CommonTimingLinkGroup (Figure 6-91) holds elements that are common to both a JourneyPatternTimingLink and a VehicleJourneyTimingLink; the JourneyPatternTimingLink instances provide default values to use on dependent VehicleJourneyTimingLink instances if no specific override is provided on a particular VehicleJourneyTimingLink.

- **JourneyPattern Ref**: Optional reference to parent JourneyPattern. Normally not stated as given by containing context, but may be specified when using timing link as a stand-alone artefact. If given by context, this value is ignored. (TXC v2.4)
- **HailAndRide**: Whether link operates as a Hail and Ride service. Normally stops at both ends of a link flagged as HailAndRide will be HailAndRide stops.
- **Express**: Whether link operates as an express section (that is, typically going past a stop without stopping at one or both ends of the link).
- **StoppingArrangements**: Text description of facilities/requirements for stopping associated with link.
- **DutyCrewCode**: Code identifying duty crew operating bus over link. Note that if used, a value need not be specified on every link of a journey pattern: any value specified is assumed to run for all intervening links until the next link with a value is encountered. (NB this element should more correctly be called **DutyCrewRef**).

- **CommercialBasis**: on which this link of the service is offered. See for allowed values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contracted</td>
<td>Service on Link is contracted</td>
</tr>
<tr>
<td>notContracted</td>
<td>Service on Link is not contracted.</td>
</tr>
<tr>
<td>inherit</td>
<td>Basis is same as parent vehicle Journey or Journey Pattern</td>
</tr>
<tr>
<td>unknown</td>
<td>Basis is unknown.</td>
</tr>
</tbody>
</table>

**Table 6-46 – Allowed Values for TimingLink / CommercialBasis**

---

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.

**Figure 6-90 – JourneyPatternTimingLink / CommonTimingLinkGroup**

The **JourneyPatternTimingLinkGroup** (Figure 6-91) holds elements that are specific to a **JourneyPatternTimingLink**:

- **From**: Default usage details of from stop, specified by a **JourneyPatternStopUsageStructure**. See later.

- **To**: Default usage details of from stop, specified by a **JourneyPatternStopUsageStructure** element. See later.

- **RouteLinkRef**: Optional reference to a **RouteLink** onto which timing link projects.

- **Direction**: Direction of link. Default is ‘inherit’. See Table 6-47.

- **RunTime**: Time taken to traverse link. Normally this will be greater than zero. A **RunTime** value should only be zero, i.e. PT0M, where stops are very close together and the time between them would be less than a minute (even then it is desirable to give the value in seconds if known, e.g. PT0M35S). A journey pattern should never have zero minute timings for all its links.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>inherit</td>
<td>Use value from Journey Pattern.</td>
</tr>
<tr>
<td>inbound</td>
<td>Inbound Direction.</td>
</tr>
</tbody>
</table>
Table 6-47 – Allowed Values for VehicleJourney / Direction

- **Distance**: Distance along link path in metres.

<table>
<thead>
<tr>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>outbound</td>
</tr>
<tr>
<td>clockwise</td>
</tr>
<tr>
<td>anticlockwise</td>
</tr>
</tbody>
</table>

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.

```plaintext
Figure 6-91 – JourneyPatternTimingLinkGroup
```

6.7.8 **JourneyPatternStopUsageStructure**

The **JourneyPatternStopUsageStructure** (Figure 6-92) describes the use of a stop by the start or end of a **JourneyPatternTimingLink**, or unordered stop reference in a FlexibleJourneyPattern. It provides default values that will be inherited by the corresponding **VehicleJourneyStopUsage** elements of dependent vehicle journeys.

Both **JourneyPatternStopUsage** and **VehicleJourneyStopUsage** instances can be identified by a unique **id** attribute, and may also have a **SequenceNumber** attribute to indicate the preferred ordering of stops when presenting schedules in matrix timetable formats.

**JourneyPatternStopUsage** comprises a number of elements falling into two groups:

1. **JourneyStopUsage** Group: Shared elements common to journey pattern stop usage elements, and to vehicle journey stop usage elements.
2. **JourneyPatternStopUsage** Group: Elements specific to journey pattern stop usage elements.
The **JourneyPatternStopUsageGroup** (Figure 6-93) holds elements that are common to both a **JourneyPatternStopUsage** and a **VehicleJourneyStopUsage**. 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; the wait time is the part of the **Overall Wait Time** at the stop that has been ascribed to end of the link represented by the stop usage. When calculating departure times for a specific vehicle journey, the timing link **WaitTime** values from the respective stop use ends of the incoming and outgoing links are added together to create the total wait time at the stop. If not specified, assume zero.

- **Activity**: Activity undertaken by vehicle at stop. See Table 6-48. Defaults to **pick up and set down**.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pickUp</td>
<td>Pick up passengers.</td>
</tr>
<tr>
<td>setDown</td>
<td>Set down passengers.</td>
</tr>
<tr>
<td>pickUpAndSetDown</td>
<td>Pick up and set down passengers.</td>
</tr>
<tr>
<td>pass</td>
<td>Do not stop at stop.</td>
</tr>
</tbody>
</table>

**Table 6-48 – Allowed Values for Activity**

- **DynamicDestinationDisplay**: Journey destination applicable to vehicle at referenced stop.
- **Vias**: Names of intermediate points applicable to vehicle at referenced stop. If not specified on a VehicleJourney, value will be defaulted from JourneyPatternStopUsage. (+TXC v2.4)
  - **ViaName**: Name of an intermediate point on the journey to be shown as a via point.
  - **None**: There are no Vias to be shown
- **VariableStopAllocation**: In bus stations, bays may be allocated to a service variously on different days. This can be specified using the **VariableStopAllocation** element. See below.
- **StopOnlyOnRequest**: Whether stop is only a request stop on this journey. Default false.
- **Note**: Descriptive text note associated with stop.
- **StopAccessibility**: Accessibility details for stop – applicable for specific visit of service (TXC V2.5).

```
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. See User Guide for rules for combining wait times.

- **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.

- **Vias**: Intermediate places to be show as via points on journey. (TXC V2.4)

- **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.

- **StopAccessibility**: Stop accessibility details

```

**Figure 6-93 – JourneyPattern / JourneyStopUsageGroup**

6.7.8.2 JourneyPatternStopUsage / JourneyPatternStopUsageGroup

The **JourneyPatternStopUsageGroup** (Figure 6-94) holds information specific to a **JourneyPatternStopUsage**:

- **StopPointRef**: NaPTAN Stop at which timing link starts or ends.
- **TimingStatus**: Classification of the role of the stop as a timing point used by the journey pattern. See Table 6-49. Overrides the classification defined by the stop in NaPTAN.

<table>
<thead>
<tr>
<th>Value</th>
<th>Long Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTP</td>
<td>principalTimingPoint</td>
<td>Principal and time info point.</td>
</tr>
<tr>
<td>TIP</td>
<td>timeInfoPoint</td>
<td>Time Info Point.</td>
</tr>
<tr>
<td>OTH</td>
<td>otherPoint</td>
<td>Other Bus Stop.</td>
</tr>
</tbody>
</table>

Table 6-49 – Allowed Values for TimingStatus

- **FareStageNumber**: The fare stage number for the referenced stop. A fare stage number should be specified if the fare stage is different from that on the previous link.
- **FareStage**: Whether a fare stage is encountered while traversing the end of the timing link. This should correspond to the value implied by the **FareStageNumber**. If the two are in conflict, then the **FareStageNumber** will be assumed correct.
The **VariableStopAllocations** element (Figure 6-95) describes the variable allocation of bays in a bus station. It can be used to assign to assign specific bays for a service on specific dates. It comprises zero or more **VariableStopAllocation** elements, each specifying an individual allocation on a given date. The time of allocation is the passing time of the service at the stop. Normally the assigning stop will be of stop type ‘BCQ’ (Bus / Coach Station Variable Bay), the assigned stops of type ‘BCT’ (Bus / Coach Station Bay).

- **DateRange**: A collection of one or more open-ended date ranges, and any number of date exceptions.
  - **StartDate**: The (inclusive) start date. If omitted, the range start is open-ended, that is, it should be interpreted as "since the beginning of the service validity period".
  - **EndDate**: The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "until end of the service validity period" (which may be indefinite).

- **VariableStopPoint**: Bay or bays to which service is allocated for the specified date (and time of the service). Normally will be a NaPTAN stop of type ‘BCT (Bus / Coach Station Bay)’. If more than one stop is specified, then bays are considered to be a pool that can be used on a first come first serve basis.
  - **StopPointRef**: NaPTAN Identifier of a StopPoint.

- **DefaultStopAllocation**: Bay or pool of bays to use if no date-specific VariableStopAllocation is applicable for a given date.

---

**Figure 6-94 – JourneyPattern / JourneyPatternStopUsageGroup**

**6.7.8.3 VariableStopAllocations Element**

---

**Figure 6-95 – JourneyPattern / VariableStopAllocation Element**
6.7.8.4 StopAccessibility Element (+TXC v2.5)

The **StopAccessibility** element (Figure 6-96) specifies the accessibility of the visit to the stop for mobility impaired users. This can be used to override the Journey level defaults if there are specific considerations for a particular stop. For example, it may be that a Vehicle is normally accessible using equipment but that the hoist cannot be used at certain stops. It comprises an overall assessment and a number of detailed criteria. It comprises:

- **A MobilityImpairedAccess**: Overall assessment of the stop for accessibility. This can be used for example to indicate accessible and inaccessible stops on maps and in journey planners. See Limitation Status (Table 6-50) for allowed values. For a topologically simple stop such as an on-street bus stop, this will typically be the same as the **WheelchairAccess** status. For complex stops such as metro and rail stations it requires an overall judgement based on the accessibility of individual platforms. For example a station which requires the use of a flight of steps to reach the main platform would be considered inaccessible.

Specific assessments:

- **WheelchairAccess**: Whether stop is accessible to wheelchair users. See Limitation Status (Table 6-50) for allowed values. Normally if there is Step free access there will be wheelchair access. However wheelchair access may additional require assistance, use of a boarding ramp etc.
- **StepFreeAccess**: Whether stop is accessible without the use of steps. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values.
- **EscalatorFreeAccess**: Whether stop is accessible without the use of escalator. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values.
- **LiftFreeAccess**: Whether stop is accessible without the use of lifts. Normally only relevant for off-street stops such as Station platforms. See Limitation Status (Table 6-50) for allowed values. Lift free access may be of concern to sufferers from claustrophobia, autism and other conditions.

Limitation Status (Table 6-50).shows the allowed values for accessibility assessments. Note that a value of unknown should be used if the accessibility is not known.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Stop is considered accessible according to criteria.</td>
</tr>
<tr>
<td>false</td>
<td>Stop is not considered accessible according to criteria.</td>
</tr>
<tr>
<td>partial</td>
<td>Stop is partial accessible according to criteria: some areas are not accessible.</td>
</tr>
<tr>
<td>unknown</td>
<td>The accessibility of the stop according to the criteria a not known.</td>
</tr>
</tbody>
</table>

Table 6-50 – Allowed Values for LimitationStatus

The Limitation Status includes an "unknown" value which can be used when the accessibility status is not known. It is reasonable to assume that Air, Bus and Coach Stops will usually be accessible even if a value is not specified. See Table 6-51.

<table>
<thead>
<tr>
<th>Mode of stop</th>
<th>Value to assume if unspecified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air</td>
<td>True</td>
</tr>
<tr>
<td>Rail</td>
<td>Unknown</td>
</tr>
<tr>
<td>Metro</td>
<td>Unknown</td>
</tr>
<tr>
<td>Ferry</td>
<td>Unknown</td>
</tr>
<tr>
<td>Tram</td>
<td>Unknown</td>
</tr>
<tr>
<td>Bus</td>
<td>True</td>
</tr>
<tr>
<td>Coach</td>
<td>True</td>
</tr>
</tbody>
</table>

Table 6-51 – Accessibility defaults by mode

Assistance values:

- **AccessVehicle**: Details on accessibility for wheelchair users. See below
- **AssistanceAvailability**: Whether there is an assistance service available for disabled users and if so how it should be booked. See Table 6-8 earlier for allowed values.
- **AssistanceTimes**: Times when assistance is available.
- **DayType**: Type of day (Monday to Sunday) and **Timeband**. See below.
- **OperatorRef**: Identifier of operator who provides service. This can be used to integrate booking details and other information.
- **InfoUrl**: Public URL with information about accessibility at the stop.

Further details
- **ServicesAtStopAreNormallyAccessible**: Whether services at the stop are normally accessible, for example the vehicle type has low floor, a wheelchair hoist, etc. This is a default value for indicative guidance only. It may be that specific services are not accessible. See Limitation Status (Table 6-50) for allowed values.
- **Note**: Any comment about the accessibility at the stop for the journey.

![StopAccessibilityStructure](image-url)

**Figure 6-96 – StopAccessibility Element**

6.7.9 JourneyPatternInterchange Element

The **JourneyPatternInterchange** element *(Figure 6-97)* describes an interchange connecting two **JourneyPatterns**. Each interchange can be identified by a unique **id** attribute, and comprises a number of elements, falling into two groups:
1. **CommonInterchangeGroup**: Shared elements common to journey pattern interchanges and vehicle journey interchanges. See below.

2. **JourneyPatternInterchangeGroup**: Elements specific to vehicle journey interchanges. See below.

### Figure 6-97 – JourneyPatternInterchange Element

#### 6.7.9.1 JourneyPatternInterchange / CommonInterchangeGroup

The **CommonInterchangeGroup** (Figure 6-98) holds information that is common to both a **JourneyPatternInterchange** and a **VehicleJourneyInterchange**.

- **MinInterchangeTime**: Minimum time to allow for changing services at the interchange.
- **MaxInterchangeTime**: Maximum time that connecting service will wait at the interchange.
- **TransferMode**: Method of transport used to make transfer between inbound and outbound journeys at the interchange. See Table 6-52.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>walk</td>
<td>Walk transfer.</td>
</tr>
<tr>
<td>bus</td>
<td>Bus transfer.</td>
</tr>
<tr>
<td>train</td>
<td>Train transfer.</td>
</tr>
<tr>
<td>tram</td>
<td>Tram transfer.</td>
</tr>
<tr>
<td>metro</td>
<td>Metro transfer.</td>
</tr>
<tr>
<td>coach</td>
<td>Coach transfer.</td>
</tr>
<tr>
<td>ferry</td>
<td>Ferry transfer.</td>
</tr>
<tr>
<td>air</td>
<td>Air transfer.</td>
</tr>
<tr>
<td>taxi</td>
<td>Taxi transfer.</td>
</tr>
<tr>
<td>cycle</td>
<td>Cycle transfer.</td>
</tr>
<tr>
<td>movingWalkway</td>
<td>Moving Walkway transfer.</td>
</tr>
</tbody>
</table>

**Table 6-52 – Allowed Values for TransferMode**

- **ValidityPeriod**: Period when the interchange is valid.
  - **StartDate**: Inclusive date of start of validity period.
  - **EndDate**: Inclusive date of end of validity period.
- **StoppingArrangements**: Text description of stopping arrangements for the interchange.
- **InterchangeActivity**: Activity taking place between incoming and outgoing **VehicleJourney** instances at an interchange. See Table 6-53.
- **InterchangeInfoGroup**: Additional information about the nature of the interchange. See below.
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.

### CommonInterchangeGroup

- **MinInterchangeTime**
  - Type: DurationType
  - Minimum time to allow to change services at the interchange. Assume normal walking speed. Default is zero.

- **MaxInterchangeTime**
  - Type: DurationType
  - Maximum time available to change services at the interchange. A duration: for guaranteed connections this may be a upper limit.

- **TransferMode**
  - Type: AllModesEnumeration
  - Method of transport available/assumed at Interchange. Enumerated value.

- **ValidityPeriod**
  - Type: HalfOpenDateRangeStructure
  - Period of Interchange validity.
    - **StartDate**
      - Type: StartDateType
      - The (inclusive) start date.
    - **EndDate**
      - Type: EndDateType
      - The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever".

- **StoppingArrangements**
  - Type: NaturalLanguageStringStructure
  - Description of facilities/requirements for stopping associated with components of a service. @lang.

- **InterchangeActivity**
  - Type: InterchangeActivityEnumeration
  - Activity taking place between incoming and outgoing Journey at an Interchange. Enumerated value. Default is Change.

- **Weighting**
  - Type: xsd:integer
  - Arbitrary relative weighting that can be associated with connection for guidance to journey planners. This is for private application use.

---

**Figure 6-98 – CommonInterchangeGroup**

6.7.9.2 JourneyPatternInterchange / InterchangeInfoGroup

The InterchangeInfoGroup (Figure 6-99) holds additional information about the nature of the interchange.

- **CrossBorder**: Whether the connection crosses a border.
- **GuaranteedConnection**: Whether the connection is guaranteed.
- **ChangeLineNumber**: Whether the service changes number at the connection.

---

**Figure 6-99 – JourneyPatternInterchange / InterchangeInfoGroup**

6.7.9.3 JourneyPatternInterchange / JourneyPatternInterchangeGroup

The JourneyPatternInterchangeGroup holds elements that are specific to a JourneyPatternInterchange, and describe the connection between two journeys.

- **Inbound**
- **JourneyPatternRef**: Incoming *JourneyPattern* that connects to the interchange.
- **StopUsageRef**: Reference to the *JourneyPatternStopUsage* of the *JourneyPatternTimingLink* that connects inbound *JourneyPattern* to the interchange.

- **Outbound**
  - **JourneyPatternRef**: On-going *JourneyPattern* that connects from the interchange.
  - **StopUsageRef**: Reference to the *JourneyPatternStopUsage* of the *JourneyPatternTimingLink* that connects the outbound *JourneyPattern* to the interchange.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>change</td>
<td>Service changes at interchange</td>
</tr>
<tr>
<td>join</td>
<td>Service joins at interchange.</td>
</tr>
<tr>
<td>split</td>
<td>Service splits at interchange.</td>
</tr>
<tr>
<td>through</td>
<td>Through journey.</td>
</tr>
</tbody>
</table>

Table 6-53 – Allowed Values for InterchangeActivity

![Diagram of JourneyPatternInterchange / JourneyPatternInterchangeGroup](Image)

Figure 6-100 – JourneyPatternInterchange / JourneyPatternInterchangeGroup
6.8 VehicleJourney Element

A VehicleJourney (Figure 6-101) describes a specific journey of a vehicle following a fixed JourneyPattern of a StandardService. The JourneyPattern comprises one or more VehicleJourneyTimingLink elements: the order of links represents the order in which they are traversed. A VehicleJourney comprises a number of elements; the elements fall into three groups:

1. CommonJourneyGroup: Shared elements common to journey patterns and vehicle journeys. See JourneyPattern / CommonJourneyGroup earlier. Allows individual properties to be overridden on a vehicle journey: if not specified the property from the journey pattern will be used. The facility to override information from the JourneyPattern in the VehicleJourney should be used sparingly. Information that is shared many times with other journeys following the same journey pattern should be added to the journey pattern and not be repeated unnecessarily.

2. VehicleJourneyGroup: Elements specific to vehicle journeys, both fixed and flexible.


### VehicleJourney

A specific scheduled journey by an individual vehicle making up a standard part of a service.

- @CreationDateTime
- @ModificationDateTime
- @Modification
- @RevisionNumber
- @Status
- @SequenceNumber.

### 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.

### VehicleJourneyGroup

Properties common to both flexible and standard vehicle journeys.

### StandardVehicleJourneyGroup

Properties specific to a standard vehicle journey.

### Extensions

Extensions to schema. (Wrapper tag used to avoid problems with handling of optional ‘any’ by some validators).

#### Figure 6-101 – VehicleJourney Element

6.8.1 VehicleJourney / VehicleJourneyGroup

The VehicleJourneyGroup (Figure 6-102): holds elements that are common to both fixed and flexible types of VehicleJourney.

- **VehicleJourneyCode**: A unique code that can be used to identify the VehicleJourney.
- **ServiceRef**: The Service to which the VehicleJourney belongs.
- **LineRef**: The Service / Line that the VehicleJourney serves.
- **Referenced Journey pattern**: One of the following:
  - **JourneyPatternRef**: The JourneyPattern over which the VehicleJourney runs. Route, timing links and other properties will be derived from the specified journey pattern.
  - **VehicleJourneyRef**: Reuse the VehicleJourneyTimingLink elements of the referenced VehicleJourney, and follow its JourneyPattern. If a VehicleJourneyRef is specified, then any VehicleJourneyTimingLink instances of the dependent VehicleJourney will be ignored.
- **StartDeadRun**: Initial “dead run” for positioning the vehicle before it traverses its timing links. See below.
- **EndDeadRun**: Final “dead run” link for positioning the vehicle after it traverses its timing links. See below.
VehicleJourneyInterchange: Interchanges where the vehicle journey connects with another vehicle journey. See later.

Note: Any additional notes on the VehicleJourney. See below.

VehicleJourneyGroup: Properties common to both flexible and standard vehicle journeys.

VehicleJourneyGroup Identifier of the vehicle journey.

VehicleJourneyCode type VehicleJourneyCodeType

Identification of the vehicle journey.

ServiceRef type ServiceRefStructure

The Service that the VehicleJourney contributes to.

LineRef type LineIdType

The Line for which the VehicleJourney operates.

JourneyPatternRef type JourneyPatternIdType

The Journey Pattern which the VehicleJourney runs.

VehicleJourneyRef type VehicleJourneyCodeType

Reuse the VehicleJourney TimingLink elements of the referenced VehicleJourney; this can give a dramatic reduction in file sizes. The Journey Pattern will be that of the referenced journey. All the links of the referenced journey are used, in the same order. Cyclic references are not allowed.

StartDeadRun type DeadRunStructure

Positioning journey to place vehicle at beginning of vehicle journey. An ordered collection of positioning links. @id.

EndDeadRun type DeadRunStructure

Positioning journey to place vehicle at end of vehicle journey. @id.

VehicleJourneyInterchange type VehicleJourneyInterchangeStructure

Information about an interchange where the vehicle journey connects with another vehicle journey.

@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.

Note type NoteStructure

Any additional notes on the VehicleJourney.

Figure 6-102 – VehicleJourney / VehicleJourneyGroup

6.8.2 VehicleJourney / StandardVehicleJourneyGroup

The StandardVehicleJourneyGroup (Figure 6-103) holds elements that are specific to fixed VehicleJourney instances:

- DepartureDayShift: Whether the DepartureTime is to be shown as the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular bed, do for example a journey that runs in at 0-0:05 that is part of the Monday to Friday service can be shown as a as expressed by the profile (+TXC v2.4). See earlier discussion of the effect of
• **Frequency:** Describes service frequency for frequency based services. See below.

• **VehicleJourneyTimingLink:** An ordered collection of timing links making up the VehicleJourney. See VehicleJourneyTimingLink later.

<table>
<thead>
<tr>
<th>StandardVehicleJourneyGroup</th>
<th>Properties specific to a standard vehicle journey.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DepartureTime</td>
<td>Time of departure from origin stop of the Vehicle Journey.</td>
</tr>
<tr>
<td>DepartureDayShift</td>
<td>Shift in day's forward or backwards if time is not on same day as daytype of operating profile. For example a 00:10 Saturday service that was part of Friday service would have DayShift + 1, A 23:50 Sunday service that was part of Monday to Friday service would have DayShift -1.</td>
</tr>
<tr>
<td>Frequency</td>
<td>Frequency details for a frequent service, that is one that runs as a shuttle rather than to a set timetable.</td>
</tr>
<tr>
<td>VehicleJourneyTimingLink</td>
<td>Timing link making up the vehicle journey. Ordered in sequence of physical traversal. For each link there should be a corresponding JourneyPatternTimingLink. @id, @creationDateTime, @modificationDateTime, @modification, @revisionNumber.</td>
</tr>
</tbody>
</table>

Figure 6-103 – VehicleJourney / StandardVehicleJourneyGroup

6.8.3 Common VehicleJourney Subelements

6.8.3.1 VehicleJourney / DeadRun Element

A **DeadRun** (Figure 6-104) models a **StartDeadRun** or **EndDeadRun**, that is, a positioning run at the start or end of a journey; it is used to place a vehicle in position to start the service, or to retrieve it at the end of the journey.

A DeadRun can be used to track the vehicle in a real time system to give better predictions for the part of the journey that is in public service, and so should be provided along with the run times that relate to the running of the vehicle during the positioning journey (i.e. as a JourneyPatternTimingLink with RunTimes). The dead run should not be treated as an out of service part of the journey in order to hide that part of the journey from the public timetables unless the vehicle actually operates out of service to the stated times.

A **DeadRun** comprises:

• **PositioningLink:** One or more links describing how the vehicle travels to or from the route. See below.

• **ShortWorking:** If the dead run intercepts the journey pattern at a point, identifies the start or end point on the journey pattern at which the interception happens. May be used even if no positioning link is specified.
  - **JourneyPatternTimingLinkRef:** Link at which journey starts or finishes.
DeadRunStructure (extension)

Dead Run definition.

DeadRun
(extension)
type
DeadRunStructure

attributes
Positioning journey step to place vehicle at beginning or end of vehicle journey.

@id,
@CreationDateTime,
@ModificationDateTime,
@Modification,
@RevisionNumber.

PositioningLink
(type
PositioningLinkStructure

ShortWorking
(type
PositioningLinkStructure

Used to truncate the start or end of a vehicle journey part way through the journey pattern. The vehicle journey inherits all Journey Pattern TimingLinks 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).

Extensions
(type
ExtensionsAnyStructure

Extensions to schema. (Wrapper tag used to avoid problems with handling of optional ‘any’ by some validators).

Figure 6-104 – VehicleJourney / DeadRun Element

6.8.3.2 VehicleJourney / PositioningLink Element

A PositioningLink (Figure 6-105) models a step of a DeadRun. It comprises:

- **RunTime**: Time taken to traverse link.
- **From**: From point, a stop, garage, or location. See PositioningLinkUsage below.
- **To**: To point; also a stop, garage, or location. See PositioningLinkUsage below.
- **DutyCrewRef**: Identifier of crew manning vehicle over link. (+TXC v2.4)
- **Track**: Path taken by vehicle when traversing the positioning link. See RouteLink / Track element earlier.
A PositioningLink (Figure 6-105) models one end of a PositioningLink. It comprises one of the following:

- **StopPointRef**: A NaPTAN stop point. Usually on the journey pattern, but can be completely arbitrary – e.g. a stop on another route from which the bus is coming.
- **GarageRef**: A Garage defined for the operator of the Service to which vehicle journey belongs.
- **LayoverPointRef**: A LayoverPoint defined for the JourneyPattern.
- **Location**: An arbitrary location specified by spatial coordinates.
6.8.3.4 VehicleJourney / Frequency Element

**Frequency** (Figure 6-107) gives details about a frequency based service, that is, one that runs as a shuttle rather than to a set timetable.

- **EndTime**: Describes when the frequency based period ends.
  
  The frequency can be specified in one of two ways:
  
  - **Interval**: Describes the expected frequency of a service in quantitative terms as an interval. See 6.8.3.5 below.
  
  - **Minutes past the hour**: Describes the expected frequency of a service in quantitative terms.
    
    Comprises:
    
    - **Minutes**: One or more times past the hour.
    
  - **FrequentService**: Formally declares the journey to be a frequent service, with an interval of at least once every 10 minutes. A minimum frequency should be specified.

6.8.3.5 Frequency Interval

**Interval** element describes the expected frequency of a service in quantitative terms as an interval, that can be used to describe the service periodicity such as “Every ten minutes”. Comprises:

- **ScheduledFrequency**: The scheduled time gap between departures.
- **MinimumFrequency**: The minimum time gap between departures.
- **MaximumFrequency**: The maximum time gap between.
- **Description**: Override description text for Frequency.
6.8.4 VehicleJourneyTimingLink Element

A VehicleJourneyTimingLink (Figure 6-109) models the link between two stops of a vehicle journey. Each VehicleJourneyTimingLink can be identified by a unique id attribute, and comprises a number of elements. The elements fall into two groups:

1. **CommonTimingLinkGroup**: Shared elements common to journey pattern timing links and vehicle journey timing links. See JourneyPatternTimingLink / CommonTimingLinkGroup earlier.

2. **VehicleJourneyTimingLinkGroup**: Elements specific to vehicle journey timing links.

![VehicleJourneyTimingLink](image)

6.8.4.1 VehicleJourneyTimingLink / VehicleJourneyTimingLinkGroup

The VehicleJourneyTimingLinkGroup (Figure 6-110) holds information is specific to a VehicleJourneyTimingLink:

- **ParentVehicleJourneyRef**: 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 already given by context, this value is ignored. (+TXC v2.4).
**JourneyPatternTimingLinkRef**: Reference to a *JourneyPatternTimingLink* onto which timing link projects, and which defines the origin and destination points of the link. See *JourneyPatternTimingLink* earlier.

**RunTime**: Time taken to traverse link. Defaults to value specified for *JourneyPatternTimingLink*.

**From**: Usage details of from stop, specified by a *VehicleJourneyStopUsage* element. This projects onto the *From / JourneyPatternStopUsage* of the corresponding *JourneyPatternTimingLink*.

**To**: Usage details of from stop, specified by a *VehicleJourneyStopUsage* element. This projects onto the *To / JourneyPatternStopUsage* of the corresponding *JourneyPatternTimingLink*.

**Properties specific to vehicle journey timing links.**

The *VehicleJourneyTimingLinkGroup* (Figure 6-110) describes the use of a stop by the start or end of a *VehicleJourneyTimingLink*. The *VehicleJourneyStopUsage* can be identified by a unique *id* attribute, and comprises:

- A *JourneyStopUsageGroup*: see *JourneyPatternStopUsage* earlier. Any values specified override the values specified for the underlying journey pattern.
- A Frequency *Interval*. See 6.8.3.5. If a value is specified, it indicates this is a Partial Frequency Based service and that the stop passing time should be presented as a frequency (e.g. "Every five minutes") rather than an absolute time (e.g. "10:23"). Note that an absolute time will still be computed in case absolute times are needed for any subsequent stops. See discussion of computing of passing times earlier. (+TXC v2.4). An override Interval will normally only be used only on the departure end of a timing link, that is for the *From / Stop UsageGroup*.

---

**Figure 6-110 – VehicleJourneyTimingLinkGroup**

6.8.5 *VehicleJourneyTimingLink / VehicleJourneyStopUsage Element*

The *VehicleJourneyStopUsageStructure* (Figure 6-94) describes the use of a stop by the start or end of a *VehicleJourneyTimingLink*. The *VehicleJourneyStopUsage* can be identified by a unique *id* attribute, and comprises:

- A *JourneyStopUsageGroup*: see *JourneyPatternStopUsage* earlier. Any values specified override the values specified for the underlying journey pattern.
- A Frequency *Interval*. See 6.8.3.5. If a value is specified, it indicates this is a Partial Frequency Based service and that the stop passing time should be presented as a frequency (e.g. "Every five minutes") rather than an absolute time (e.g. "10:23"). Note that an absolute time will still be computed in case absolute times are needed for any subsequent stops. See discussion of computing of passing times earlier. (+TXC v2.4). An override Interval will normally only be used only on the departure end of a timing link, that is for the *From / Stop UsageGroup*. 
Abstract Timing Link Stop Usage Structure (extension)

VehicleJourneyStopUsageStructure

Properties governing the use of a stop by one end of a VehicleJourney.

WaitTime

Type: DurationType

Time to wait at the referenced Stop, overriding any corresponding Journey Pattern Wait Time for the Stop Usage. If this time is the TOTAL wait time at the stop and thus is duplicated by a wait time associated with the stop point of the preceding/following link.

Activity

Type: VehicleAtStopActivityEnumeration

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

Type: NaturalLanguageStringStructure

Journey destination applicable to vehicle at referenced Stop. (TxC 2.4)

Vias

Type: ViasStructure

Intermediate places to be shown as via points on journey. (TxC 2.4)

VariableStopAllocations

Type: VariableStopAllocationsStructure

For bus stations where allocation of bays 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

Type: NotesStructure

Notes associated with this usage. TxC 2.4. Any Vehicle Journey notes completely replace any Journey Pattern notes.

StopAccessibility

Type: StopAccessibilityStructure

Stop accessibility details

Extensions

Type: ExtensionsAnyStructure

Extensions to schema. (Wrapper tag used to avoid problems with handling of optional ‘any’ by some validators).

Figure 6-111 – VehicleJourneyStopUsage Element

6.8.6 VehicleJourney / VehicleJourneyInterchange Element

The VehicleJourneyInterchange element (Figure 6-112) records information about an interchange at which the vehicle journey connects with another vehicle journey. Each interchange can be identified by a unique id attribute, and comprises a number of elements, falling into two groups:


VehicleJourneyInterchange

type | VehicleJourneyInterchangeStructure

Information about an interchange where the vehicle journey connects with another vehicle journey.
- @id
- @CreationDateTime
- @ModificationDateTime
- @Modification
- @RevisionNumber.

VehicleJourneyInterchangeGroup

Interchange properties specific to vehicle journey interchanges.

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.

VehicleJourneyInterchangeGroup

Interchange properties specific to vehicle journey interchanges.

Extensions

type | ExtensionsAnyStructure

Extensions to schema. (Wrapper tag used to avoid problems with handling of optional 'any' by some validators).

Figure 6-112 – VehicleJourneyInterchange Element

6.8.6.1 VehicleJourney/ VehicleJourneyInterchangeGroup

The VehicleJourneyInterchangeGroup (Figure 6-113) holds elements that are specific to a VehicleJourneyInterchange:

- **VehicleJourneyRef**: Optional reference to parent VehicleJourney. Normally not stated as given by containing context, but may be specified when using the Interchange as a stand-alone artefact. If already given by context, this value is ignored, (+TXC v2.4).
- **JourneyPatternInterchangeRef**: The JourneyPatternInterchange to which this VehicleJourneyInterchange corresponds.
- **InboundVehicleJourneyPatternRef**: The VehicleJourney of the incoming journey that connects at the interchange.
- **OutboundVehicleJourneyPatternRef**: The VehicleJourney of the on-going journey that connects at the interchange.

Figure 6-113 – VehicleJourneyInterchangeGroup
6.9 **FlexibleService, FlexibleJourneyPattern, FlexibleVehicleJourney**

6.9.1 **FlexibleService Element**

The **FlexibleService** element

![Diagram of FlexibleService Element](image)

Figure 6-114 describes the flexibly routed component of a **Service**, using one or more instances.

- **ServicePartGroup** – elements common to standard and flexible journey parts. See earlier.
- **FlexibleServiceGroup** – elements specific to **FlexibleService** part.
  - **FlexibleJourneyPattern**: One or more **FlexibleJourneyPattern** elements representing the working of the service. See below.
  - **Cancellation**: If registration is a cancellation, **FlexibleJourneyPattern** may be omitted.

6.9.1.1 **FlexibleJourneyPattern Element**

The **FlexibleJourneyPattern** element (Figure 6-116) describes the availability of a flexibly routed journey of a **Service**. It is made up of two parts:
1. **CommonJourneyGroup**: Shared elements common to journey patterns and fixed and flexible vehicle journeys. See *JourneyPattern / CommonJourneyGroup* earlier. The *JourneyPattern* instances provide default values to use on dependent *FlexibleVehicleJourney* instances if no specific override is provided on an individual *FlexibleVehicleJourney*.

2. **FlexibleJourneyPatternGroup**: Elements specific to flexible journey patterns.

### 6.9.1.2 FlexibleJourneyPattern / FlexibleJourneyPatternGroup

The *FlexibleJourneyPatternGroup* (*Figure 6-116*) holds elements specific to a flexible journey pattern that describes the area of flexible operation and comprises as follows:

- **FlexibleZones**: Describes the zones that the service covers. See *FlexibleStopUsage* below.
- **FixedStopPoints**: Describes any fixed stops that can be visited by the service. See *FixedStopUsage* below.
- **BookingArrangements**: Arrangements for booking the service. See below.

### 6.9.2 FlexibleService Subelements

#### 6.9.2.1 FlexibleService / FlexibleZones Element

A flexible journey pattern describes the areas and stops covered by a flexible service as two lists: one of flexible zones, and one of fixed stops (*Figure 6-117*).

- **FlexibleZones**: Comprises a collection of *FlexibleStopUsage* instances: each is a *FlexibleStopUsage* instance with an activity (e.g. pick up, set down), and a reference to a NaPTAN stop of type *FlexibleZone*.
  - **Activity**: Activity undertaken by vehicle at stop. See *Table 6-48*. Defaults to pick up and set down.
- **StopPointRef**: NaPTAN Stop at which timing link starts or ends.

![Diagram](image)

**Figure 6-117 – FlexibleZones Element**

### 6.9.2.2 FlexibleService / FixedStopPoints Element

The **FixedStopPoints**: Figure 6-118) is an ordered collection of **FixedStopUsage** instances: each is a **JourneyPatternStopUsageStructure** (see earlier) instance with an activity (e.g. pick up, set down), and a reference to a NaPTAN fixed stop, i.e. of any type such as **MarkedPoint**, other than **FlexibleZone**.

![Diagram](image)

**Figure 6-118 – FlexibleService / FixedStopPoints Element**

### 6.9.2.3 FlexibleVehicleJourneyGroup / BookingArrangements Element

The **BookingArrangements** element (Figure 6-119) describes the booking arrangements for the flexible service:

- **Description**: Text description of booking process.
- **Phone**: Phone number by which to make bookings. See **TelephoneContactStructure**.
- **Email**: Email address to which to make bookings.
- **Address**: Postal address by which to make bookings. See **PostalAddressStructure**.
- **WebAddress**: URL of online web site by which make bookings.
- **AllBookingsTaken**: Whether all bookings are taken. Default is true.
FlexibleVehicleJourney Element

The **FlexibleVehicleJourney** element (Figure 6-120) describes the availability of a flexible journey. It adds time information to a **FlexibleJourneyPattern** instance. A **FlexibleVehicleJourney** comprises a number of elements; the elements fall into three groups:

1. **CommonJourneyGroup**: Shared elements common to journey patterns and vehicle journeys (See JourneyPattern / CommonJourneyGroup earlier).
2. **VehicleJourneyGroup**: Elements specific to both fixed and flexible vehicle journeys (See VehicleJourney / VehicleJourneyGroup earlier).
6.9.3.1 FlexibleVehicleJourneyGroup / FlexibleServiceTimes Element

The **FlexibleServiceTimes** element (Figure 6-121) describes the operational days of the service. **FlexibleServiceTimes** may either be:

- **AllDayService**: Indicating the service runs all day, or
- **PeriodsOfOperation**: A collection of at least one **ServicePeriod** element, made up of:
  - **StartTime**: Time at which time band starts.
  - **EndTime**: Time at which time band ends.

![Figure 6-121 – FlexibleVehicleJourney / FlexibleServiceTimes Element](image-url)
6.10 ConnectingVehicleJourney Element

A ConnectingVehicleJourney (Figure 6-122) describes a connecting journey. It allows a subset of a vehicle journey’s properties to be specified. A ConnectingVehicleJourney comprises a number of elements; the elements fall into two groups:

4. **CommonJourneyGroup**: Shared elements common to journey patterns and vehicle journeys. See JourneyPattern / CommonJourneyGroup earlier. Allows individual properties to be overridden on a vehicle journey: if not specified the property from the journey pattern will be used.

5. **ConnectingVehicleJourney**: Elements specific to connecting vehicle journeys, both fixed and flexible.

**Figure 6-122 – ConnectingVehicleJourney Element**

6.10.1 VehicleJourney / ConnectingVehicleJourney Group

The ConnectingVehicleJourneyGroup (Figure 6-103) holds elements that are specific to ConnectingVehicleJourneyGroup instances:

- **VehicleJourneyCode**: A unique code that can be used to identify the VehicleJourney.
- **ServiceRef**: The Service to which the VehicleJourney belongs.
- **RegistrationRef**: Reference to a registration of a journey
- **AnnotatedOperatorRef**: Reference to an operator. of journey, includes name
- **AnnotatedLineRef**: Reference to the Line that the VehicleJourney serves. Includes LineName the Public identifier of the Line.
- **ConnectingTimesGroup**: Describes the times for a ConnectingVehicleJourney. See below.
- **VehicleJourneyInterchange**: Interchanges where the vehicle journey connects with another vehicle journey.
- **Note**: Any additional notes on the VehicleJourney.
Properties specific to a standard vehicle journey.

**ConnectingVehicleJourneyGroup**

- VehicleJourneyCode
  - Type: VehicleJourneyCodeType
  - Identifies the connecting vehicle journey.
- ServiceRef
  - Type: ServiceNumberType
  - The service that the VehicleJourney contributes to.
- RegistrationRef
  - Type: VosaRegistrationNumberStructure
  - Reference to a registration.
- AnnotatedOperatorRef
  - Type: AnnotatedOperatorRefStructure
  - Reference to an operator.
- AnnotatedLineRef
  - Type: AnnotatedLineRefStructure
  - Reference to a line.
- ConnectingTimeGroup
  - Properties specific to a standard vehicle journey.
- VehicleJourneyInterchange
  - Type: VehicleJourneyInterchangeStructure
  - Information about an interchange where the vehicle journey connects with another vehicle journey.
- Note
  - Type: NoteStructure
  - Any additional notes on the VehicleJourney.

![Diagram of ConnectingVehicleJourney Element](image)

**Figure 6-123 – ConnectingVehicleJourney Element**

In this section we describe the schema elements used to specify operational dates and times in TransXChange. These are common to both Flexible and Standard services. See also the earlier section 3.17.8 on Modelling Operational days, which sets out the rules used for combining the various day type and date elements.

6.10.1.1 ConnectingVehicleJourney / ConnectingTimesGroup

The **ConnectingTimesGroup** (Figure 6-124) describes the times for a **ConnectingVehicleJourney**.

- **ArrivalTime**: Time of arrival at destination stop of the VehicleJourney. Used for feeder journeys.
- **ArrivalDayShift**: Whether the ArrivalTime is the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular day type as expressed by the profile.
- **DepartureTime**: Time of departure from origin stop of the VehicleJourney.
- **DepartureDayShift**: Whether the DepartureTime is the same day (0 or omit,) the next day (+1) or the Previous day. (-1) This allows Journeys that start or end after midnight to be included in a particular day type as expressed by the profile.
- **Frequency**: Describes service frequency for frequency based services. See below.
Properties specific to a standard vehicle journey.

**ConnectingTimeGroup**

- **ArrivalTime**
  - Type: xsd:time
  - Time of departure from origin stop of the Vehicle Journey.

- **ArrivalDayShift**
  - Type: xsd:integer
  - Shift in days forward or backwards if time is not on same day as day type of operating profile.

- **DepartureTime**
  - Type: xsd:time
  - Time of departure from origin stop of the Vehicle Journey.

- **DepartureDayShift**
  - Type: xsd:integer
  - Shift in days forward or backwards if time is not on same day as day type of operating profile.

- **Frequency**
  - Type: FrequentServiceStructure
  - Frequency details for a frequent service, that is one that runs as a shuttle rather than to a set timetable.

**Figure 6-124 – ConnectingVehicleJourney / ConnectingTimesGroup**

### 6.11 OperatingProfile Element

The **OperatingProfile** element (Figure 6-125) specifies on which days a service operates. An **OperatingProfile** can be specified on both a **VehicleJourney**, a **JourneyPattern** and on a **Service**; the **VehicleJourney** values override those of the **JourneyPattern** or **Service**. It is made up of two groups:

1. **Normal** operating profile group: describes normal regular behaviour.
2. **Special** operating profile group: describes behaviour on bank holidays and other exceptional days.

If no **OperatingProfile** profile is specified a default value will be used. *This assumes the service runs Monday to Friday every day of the year.* This default will be shown by the Publisher on the **Registration** particulars and Matrix footnote. (TXC v2.4)

#### 6.11.1 Normal OperatingProfileGroup

The **OperatingProfile** normal elements describe the regular operation of the service and comprise the following elements:

- **RegularDayType**: specifies the days on which the service normally runs. See below. Defaults to **MondayToSunday**.
- **PeriodicDayType**: qualifies the **RegularDayType** days with any specific weeks of the month that the service runs. It is logically ‘ANDed’ with **RegularDayType**, so that you may specify for example ‘Wednesdays, first and third weeks of the month’.
- **ServicedOrganisationDayType**: Specifies that the service runs or does not run on the working days or holidays of a nominated organisation such as a school or Local Education Authority. See **ServicedOrganisation** days below. **ServicedOrganisationDayType** is ‘ANDed’ with **RegularDayType** and any **PeriodicDayType** values.
6.11.2 Special OperatingProfileGroup

The **OperatingProfile special** elements describe exceptions to the normal days of operation and comprise the following elements:

- **SpecialDaysOperation**: Describes the specific dates (other than standard bank holiday types) when the service will operate differently from its normal service. **DaysOfOperation** and **DaysOfNonOperation** can be specified separately. See below.

- **BankHolidayOperation**: Describes how the service will operate on bank holidays. **DaysOfOperation** and **DaysOfNonOperation** can be specified separately. See below.

---

**Figure 6-125 – OperatingProfile Element**

6.11.3 OperatingProfile Subelements

6.11.3.1 OperatingProfile / RegularDayType Element

The **RegularDayType** element (Figure 6-126) specifies the normal days of operation of the associated service, journey pattern or vehicle journey. It comprises either:

- **DaysOfWeek**: Week days on which service operates. See below.

- **HolidaysOnly**: Service only runs on holidays specified by **OperatingProfile special** elements.

---

**Figure 6-126 – OperatingProfile / RegularDayType Element**

6.11.3.2 OperatingProfile / RegularDayType / DaysOfWeek Element

The **DaysOfWeek** element specifies any combination of day types using a **DayGroup** structure (Figure 6-127). It allows any meaningful combination of the following:

- Week days:
  - Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday

- Groups of days:
  - MondayToFriday, MondayToSaturday, MondayToSunday, NotSaturday, Weekend: Saturday and Sunday

---
The PeriodicDayType / WeekOfMonth element (Figure 6-128) specifies any combination of week types within a month, using up to four WeekNumber elements, i.e. any subset of four elements out of the set of numbers 1, 2, 3, 4, 5. The week numbers are combined with the day type, for example: ‘First Wednesday in the month’.

6.11.3.4 SpecialDaysOperation Element: DaysOfOperation, DaysOfNonOperation

The SpecialDaysOperation element (Figure 6-129) describes specific dates when a service does or does not operate (other than Bank Holiday day types), and comprises two collections of DateRange elements, wrapped in DaysOfOperation and DaysOfNonOperation elements respectively. If conflicting dates are specified, days of non-operation are given precedence.
6.11.3.5 DateRange

The **DateRange** element (Figure 6-130) describes a period. Each range is specified with the following:

- **StartDate**: Inclusive date on which period starts.
- **EndDate**: Inclusive date on which period ends.
- **Note**: Annotation about period.

6.11.3.6 OperatingProfile / BankHolidayOperation

The **BankHolidayOperation** element (Figure 6-131) describes how the service does or does not operate on bank holidays, and comprises two collections of **BankHolidayStructure** elements, wrapped in **DaysOfOperation** and **DaysOfNonOperation** elements respectively. If conflicting dates are specified, days of non-operation are given precedence.
6.11.3.7 OperatingProfile / BankHoliday Elements

Holiday day types are explicitly enumerated using the BankHolidayOperationStructure (see Figure 6-132), which allows individual holidays or combinations of holidays to be enumerated. The AllBankHolidaysGroup is used to denote all Bank Holidays in the country in which the service runs.

- EarlyRunOffGroup can be used to indicate special services for Christmas and New Year’s Eve.
  - EarlyRunOffDays: any Early Run off Day
  - ChristmasEve, NewYearsEve: Specific early run off days.

6.11.3.8 OperatingProfile / AllBankHolidayGroup

The AllBankHolidaysGroup specifies standard and custom bank holidays.

- A special element AllBankHolidays is used to denote all standard Bank Holidays in the country in which the service runs. See Table 6-54.
  - The HolidayMondays element can be used to denote all the summer Bank holiday Mondays.
- OtherBankHoliday can be used to specify additional special holidays.
  - Description: specifies the name of the element that can be used to denote all the summer Bank holiday Mondays.
  - Date: specifies the date on which the holiday occurs.

A special element AllBankHolidays is used to denote all standard Bank Holidays in the country in which the service runs. See Figure 6-133 and Table 6-54.
- The *HolidayMondays* element can be used to denote all the summer Bank holiday Mondays.
- *Christmas* can be used to indicate special services for actual *ChristmasDay* (strictly the 25th December) and *BoxingDay* (strictly the 26th December).
- The *HolidayMondays* element can be used to denote all the summer Bank holiday Mondays.
- The *AllHolidaysExceptChristmas* element can be used to denote all the Bank holidays in the year except for *ChristmasDay* and *BoxingDay*.
- *DisplacementHolidays* can be used to indicate special services for Public holidays that are awarded when calendar based holidays such as Christmas Day, Boxing Day or New Year’s Eve fall at a weekend so a compensating weekday, usually a Monday or Friday, is also made a public holiday. Sometimes different timetables are used for the Displacement Holiday from those that would be used for the actual day itself.

<table>
<thead>
<tr>
<th>Group</th>
<th>Subgroup</th>
<th>England &amp; Wales</th>
<th>Scotland</th>
<th>MM/DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllBankHolidays</td>
<td>AllHolidays Except Christmas</td>
<td>NewYearsDay</td>
<td>NewYearsDay</td>
<td>01/01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jan2ndScotland</td>
<td>01/02</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Holiday Mondays</td>
<td>GoodFriday</td>
<td>GoodFriday</td>
<td>var</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StAndrewsDay</td>
<td>11/30</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>Christmas</td>
<td>ChristmasDay</td>
<td>ChristmasDay</td>
<td>12/25</td>
</tr>
<tr>
<td></td>
<td>BoxingDay</td>
<td>BoxingDay</td>
<td>12/26</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Displacement Holidays</td>
<td>ChristmasDayHoliday</td>
<td>ChristmasDayHoliday</td>
<td>var</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BoxingDayHoliday</td>
<td>BoxingDayHoliday</td>
<td>var</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NewYearsDayHoliday</td>
<td>NewYearsDayHoliday</td>
<td>var</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jan2ndScotlandHoliday</td>
<td>Jan2ndScotlandHoliday</td>
<td>var</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StAndrewsDayHoliday</td>
<td>StAndrewsDayHoliday</td>
<td>var</td>
</tr>
<tr>
<td>EarlyRunOff</td>
<td>-</td>
<td>ChristmasEve</td>
<td>ChristmasEve</td>
<td>12/23</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>NewYearsEve</td>
<td>NewYearsEve</td>
<td>12/31</td>
</tr>
</tbody>
</table>

Table 6-54 – AllBankHolidays by Country

*Not official but often observed by Scottish banks and Retailers*
6.12 ServicedOrganisation Element

Operational days can also be specified in terms of the working days or holidays of specified organisations, for example schools. The ServicedOrganisation element is used to define the organisations covered, and to specify their working and non-working days.

A TransXChange document may contain a collection of ServicedOrganisation definitions. Each ServicedOrganisation definition (Figure 6-135) comprises:

- **OrganisationCode**: Identifier of the ServicedOrganisation.
- **PrivateCode**: Alternative code to support interoperability (+TXC v2.4)
- **Name**: Name of the ServicedOrganisation.
- **OrganisationContactGroup**: Contact details for organisation. See below. (+TXC v2.4)
- **ServicedOrganisationClassification**: Indication of the nature of an organisation. See Table 6-55 below. (+TXC v2.4)
### Table 6-55 – Allowed values for Serviced Organisation Type

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>school</td>
<td>School for which services will vary if open</td>
</tr>
<tr>
<td>office</td>
<td>Office for which services will vary if open</td>
</tr>
<tr>
<td>retailSite</td>
<td>Mall or major shop for which services will vary if open</td>
</tr>
<tr>
<td>touristAttraction</td>
<td>POI or tourist attraction for which services will vary if open</td>
</tr>
<tr>
<td>market</td>
<td>Street market for which services will vary if open</td>
</tr>
<tr>
<td>factory</td>
<td>Factory or Works for which services will vary if open</td>
</tr>
<tr>
<td>college</td>
<td>College or University for which services will vary if open</td>
</tr>
<tr>
<td>military</td>
<td>Military base for which services will vary if open</td>
</tr>
</tbody>
</table>

- **ServicedOrganisationDaysGroup**: Calendar for the Organisation
  - **WorkingDays**: The working days of the ServicedOrganisation, for example a LEA’s terms.
  - **Holidays**: The non-working days of the ServicedOrganisation, for example a LEA’s holidays.
- **ParentServicedOrganisationRef**: Identifier of another ServicedOrganisation that is the element’s parent. References should be acyclic. Working days and holidays specified for a parent are used as defaults for all child organisations, unless specifically overridden on the child instance.
- **LocalityGroup**: a calendar for the Organisation
  - **NptgLocalityRef**: The NPTG Locality with which the ServicedOrganisation is associated

![Figure 6-135 – ServicedOrganisation Element](image-url)
6.12.1 ServicedOrganisation Subelements

6.12.1.1 ServicedOrganisation / DatePattern Element

The DatePattern element (Figure 6-136) specifies a group of one or more non-contiguous periods as a collection of date ranges. See Modelling operation days for precedence of overlapping dates.

- **DateRange**: A collection of one or more open-ended date ranges, and any number of date exceptions.
  - **StartDate**: The (inclusive) start date. If omitted, the range start is open-ended, that is, it should be interpreted as "since the beginning of time".
  - **EndDate**: The (inclusive) end date. If omitted, the range end is open-ended, that is, it should be interpreted as "forever". If it is the same as the Start date it specifies a single day.
- **Description**: A description or name for the period. E.g., "Easter Term". (+TXC v2.4)
- **Provisional**: Whether the date is provisional or firm. (+TXC v2.4)
- **DateExclusion**: Individual dates within the period which should be omitted. (- TXC v2.4 THIS ELEMENT SHOULD NO LONGER BE USED: Simply use another Date Range instead).
  - **DateClassification**: A classification of the DateRange. (+TXC v2.4). See Table 6-56.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>term</td>
<td>Term time of an educational establishment.</td>
</tr>
<tr>
<td>working</td>
<td>Working day of an organisation.</td>
</tr>
<tr>
<td>inset</td>
<td>Inset day of an educational establishment.</td>
</tr>
<tr>
<td>holiday</td>
<td>Non-working day of any organisation.</td>
</tr>
<tr>
<td>other</td>
<td>Other day.</td>
</tr>
</tbody>
</table>

Table 6-56 – Allowed values for DateClassification

© Crown Copyright 2001-2014
6.13 Calendar Element (TXC v2.4)

The Calendar element provides a list of dates on which specific holiday day types occur. This can be used to resolve an undated vehicle journey to an actual date (+TXC v2.4).

A TransXChange document may contain a collection of ServicedOrganisation definitions. Each Calendar (Figure 6-137) comprises:

- **id**: Identifier of the Calendar.
- **CalendarPeriod**: Period within which the Calendar applies.
  - **StartDate**: Start date of Calendar.
  - **EndDate**: End date of Calendar.
- **Name**: Name of Calendar.
- **OperatingDays**: One or more OperatingDays contained in the Calendar.
  - **OperatingDay**: OperatingDay contained in Calendar. See below.
- **DayTypes**: One or more DayType a contained in the Calendar.
  - **DayType**: Day type defined by Calendar. See below.
The Calendar / OperatingDay element (Figure 6-138) specifies an actual calendar day and assignments it to a day type using a DayAssignment. See Modelling operation days for precedence of overlapping dates.

- **Date**: Calendar Date of Operating day.
- **StartTime**: The (inclusive) start time. If omitted, the day is assumed to start at 00:00.
- **Duration**: The length of the day. If omitted the day is assumed to be 24 hours long.
- **Day Assignment**: Assignment of Day to an Individual day type.
- **Day Identifier**: specified either as
  - **BankHolidayName**: name of a predefined Bank Holiday. (See BankHolidays in 6.11.3)
  - **PublicHoliday**: name of an arbitrary Public Holiday.
- **DayTypeRef**: Day type to which to assign OperatingDay
The Calendar / DayType element (Figure 6-138) specifies an actual day type. It reuses the same property of day elements of an OperatingProfile element. See earlier

- **Id**: Calendar Date of Operating day.
- **NormalDays**: See OperatingProfile earlier.
- **SpecialDays**: See OperatingProfile earlier.

The **SupportingDocument** element (Figure 6-140) Associates any supporting documents associated with the whole TransXChange schedule document – other documents, for example a schematic map, may be associated with individual elements using specific tags. Documents may be in any file format and are identified by a **DocumentUri**. Note that documents can also be associated more specifically with an individual **Registration**. **Registration** / **SupportingDocument** should be used in preference to this element for documents associated with a **Registration**.
A document supporting the associated element. Associated element may be (a) a specific individual registration, or (b) the whole document including all components.

**Figure 6-140 – SupportingDocument Element**
7 COMMON SCHEMA ELEMENTS

Some elements and types are common to a number of different elements in the TransXChange & NaPTAN schemas. These are described here.

7.1 LocationStructure

The LocationStructure type (Figure 7-1) is used to describe the spatial position of a stop or other point, for example on a Location element. Coordinates may be specified in Grid or WGS84 formats, or both. The primary coordinates used can be indicated by the LocationSystem value (Grid or WGS84) specified on TransXChange document root elements. Coordinates must be supplied for all elements in the specified primary coordinates, and may optionally be provided in the other system as well. NaPTAN data should be submitted in Grid format. NaPTAN data will normally be distributed in both formats.

If Grid coordinates are provided:
- **GridType**: Nominated grid system e.g. UKOS, ITM, IrishOS; UKOS is assumed by default.
- **Easting**: Easting grid coordinates of stop.
- **Northing**: Northing grid coordinates of stop.

If WGS84 coordinates are provided:
- **Longitude**: Longitude of stop in WGS84 coordinates.
- **Latitude**: Latitude of stop in WGS84 coordinates.

If Both Grid & WGS84 coordinates are provided:
- **Translation**, containing both of the above coordinate groups.

7.2 Duration Simple Type

The Duration simple type is used by a number of elements to specify a relative time in minutes and seconds. It uses a standard W3C duration type.


Durations are encoded in the format `PT99M88S`, where 99 is the minutes and 88 is the seconds. For example, ‘PT12M22S’ denotes twelve minutes and twenty-two seconds. The seconds may be
omitted for whole minutes, for example, \( PT5M \). Note that the W3C format also allows years, month, week and day intervals as well but these are not needed for timing intervals. The W3C definition allows arbitrary integer values for the minutes and arbitrary decimal values for the number of seconds can include decimal digits to arbitrary precision, thus \( PT1201M \), \( PT360.25S \) or \( PT1000S \) are valid (i.e. seconds do not have to be modulo sixty). Either seconds or minutes or both may be coded. Units may be combined in an arbitrary manner for example, \( P5M \), \( PT300S \) and \( PT3M120S \) are all valid equivalent encodings of 5 minutes.

7.3 TelephoneContactStructure Element

The **TelephoneContactStructure** (Figure 7-2) element specifies a phone number:

- **TelNationalNumber**: Full telephone number including STD prefix
- **TelExtensionNumber**: Any extension number.
- **TelCountryCode**: International country code for telephone. E.g. +44.

![Figure 7-2 – TelephoneContactStructure](image)

7.4 PostalAddressStructure Element

The **PostalAddressStructure** (Figure 7-3) element specifies a postal address. It is based on BS7666.

- **Line**: Between two (minimum) and five (maximum) lines of address. Each line must be between one character and thirty five characters long. Characters are restricted to those shown in *Table 7-1*
• **PostCode**: Post code of address. Valid UK Post code.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-z, A-Z</td>
<td>Upper or lower case alphabetic letters</td>
</tr>
<tr>
<td>0-9</td>
<td>Digits.</td>
</tr>
<tr>
<td>space</td>
<td>Space</td>
</tr>
<tr>
<td>“</td>
<td>Quote</td>
</tr>
<tr>
<td>‘</td>
<td>Apostrophe</td>
</tr>
<tr>
<td>,</td>
<td>Comma</td>
</tr>
<tr>
<td>:</td>
<td>Full stop</td>
</tr>
<tr>
<td>:</td>
<td>Colon</td>
</tr>
<tr>
<td>*</td>
<td>Asterisk</td>
</tr>
<tr>
<td>+</td>
<td>Plus</td>
</tr>
<tr>
<td>-</td>
<td>Hyphen</td>
</tr>
<tr>
<td>_</td>
<td>Underscore</td>
</tr>
<tr>
<td>!</td>
<td>Exclamation mark</td>
</tr>
<tr>
<td>?</td>
<td>Question mark</td>
</tr>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>&lt;</td>
<td>Left Angle Bracket</td>
</tr>
</tbody>
</table>

Table 7-1 – Allowed characters for Address Line

7.5 **Note Element**

A Note (Figure 7-4) models a set of notes attached to an element:

- **NoteCode**: Note identifier.
- **NoteText**: Text of note.

A Note may be used on Service, VehicleJourney & ConnectingVehicleJourney elements.
7.6 Submode Element

A Submode (Figure 7-4) describes a further classification of a mode: It must be one of nine values corresponding to the primary mode of the service: *AirSubmode, BusSubmode, CoachSubmode, FunicularSubmode, MetroSubmode, RailSubmode, TelecabinSubmode, TramSubmode, WaterSubmode*. See JourneyWeb specification for allowed values for each enumeration.

*Note Element*

Any additional notes on the VehicleJourney.

*NoteStructure*

- **NoteCode**
  - type: NoteCodeType
  - Reference code for Note.
- **NoteText**
  - type: NaturalLanguageStringStructure
  - Descriptive text of Note. @lang.

**Figure 7-4 – Note Element**

**Submode Element**

- **AirSubmode**
  - type: AirSubmodesEnumeration
  - TPEG pt08 Air submodes.
- **BusSubmode**
  - type: BusSubmodesEnumeration
  - TPEG pt05 Bus submodes.
- **CoachSubmode**
  - type: CoachSubmodesEnumeration
  - TPEG pt03 Coach submodes.
- **FunicularSubmode**
  - type: FunicularSubmodesEnumeration
  - TPEG pt10 Funicular submodes.
- **MetroSubmode**
  - type: MetroSubmodesEnumeration
  - TPEG pt04 Metro submodes.
- **TramSubmode**
  - type: TramSubmodesEnumeration
  - TPEG pt06 Tram submodes.
- **TelecabinSubmode**
  - type: TelecabinSubmodesEnumeration
  - TPEG pt09 Telecabin submodes.
- **RailSubmode**
  - type: RailSubmodesEnumeration
  - TPEG pt02 Rail submodes loc13.
- **WaterSubmode**
  - type: WaterSubmodesEnumeration
  - TPEG pt07 Water submodes.

**Figure 7-5 – Submode Element**
8 ELECTRONIC BUS SERVICE REGISTRATION PROCESS

This section summarises the anticipated process for registering a Bus Service using TransXChange. The proposed process is subject to confirmation by VOSA following formal testing in a demonstration. Registration includes the following steps:

8.1 Step 1: Preparation

The Transport Operator creates a proposal for a bus service and follows his normal arrangements for consulting local authorities and others as appropriate before registering the proposal.

8.2 Step 2: Encoding

The Transport Operator or its agent transfers the proposal onto a computer system. This could be either a system that handles the scheduling of operations (and which includes the capability to output the registration as TransXChange registration compliant XML document), or a simpler system that only creates TransXChange registration files. Some operators may use an agency to do this work for them – and in some areas the local authority might offer to act as an agent, particularly in respect of contract services. Each service Registration will create a separate TransXChange file – and these will be referenced using the operator’s next available registration number. Each change to a Registration likewise will carry a new sequential “version number”.

8.3 Step 3: Transmission

The Operator or the Operator’s agent logs onto the internet and connects to the VOSA Server with a normal web browser (MS Internet Explorer, Netscape, etc) using a previously-allocated username and password. The VOSA system provides a secure web connection over which the electronic registration details can be sent to the relevant Traffic Area Office. The VOSA service will offer a web page through which TransXChange files can be submitted, individually or in bulk. Files can be zipped (compressed) to reduce connection times – and multiple files can be submitted in a single zipped file. Files will be stored in a secure area of the VOSA web site and will be accessible only to the relevant Traffic Area staff, the operator making the submission and to the local authorities in whose area the service is to operate.

8.4 Step 4: Validation

The VOSA system will check that each file (unzipped, if necessary) meets the technical requirements of TransXChange – and will send a message back to the operator immediately if the file(s) fail this test. If each file passes the test, then the VOSA system will send an e-mail to the relevant local authority (or authorities) and to the operator to advise them that a registration(s) has been submitted and they can collect the submitted file(s) through their own internet connection to a secure area of the VOSA web site.

8.5 Step 5: TAN Review

Copies of the submitted file(s) are now passed into the Traffic Area Office’s business system for review in the appropriate Traffic Area Office. Some automatic checks are made on the content of the file – and the report of these checks is then passed to a case worker who will review the proposal and, once any problems are resolved, issue the acceptance for each Registration. The acceptance creates a new file – in PDF format – which provides an unchangeable record of the “registered particulars” contained in the TransXChange file. This file will be put into the secure area of the VOSA web site. Both the operator of the service, and the local authorities in whose area the service is to operate, will be advised by e-mail that the Registration has been accepted and that the PDF file of its registered particulars is available for downloading securely from the web site. If problems are found with the registration proposals during this process, the operator may be invited to make changes to their proposals and to resubmit them, starting again at Step 2 of this process.
8.6 Step 6: Acceptance and Distribution

The operator who submitted the registration (and the relevant local authorities) can then download a copy of the PDF file and can view the content of this file using freely available software (such as Adobe Acrobat Reader). This provides confirmation of the acceptance of the Registration – and sets out the only information (particularly the timetable shown only at principal “timing” points) to which the Traffic Commissioner can refer in any enforcement proceedings.

The files submitted or created during this process will remain accessible through the VOSA web site for up to 90 days using the secure access codes provided in the e-mails sent to the operator and to the relevant local authorities. After that period, the files can still be obtained on request from the relevant Traffic Area Office.

The electronic Registration process will be the same, whether the proposal is to register a new service, to change an existing registered service, or to cancel an existing service. Changes to an existing Registration require the re-submission of the complete registration details using TransXChange (but most of these details will have been stored in the operator’s systems for re-use in such circumstances). Cancellations require the submission of a very small TransXChange file that identifies the Registration concerned and the last date of operation.

TransXChange files can include timetables for use not only on normal operating days, but also those which will be used on Bank Holidays and on other special days (such as those around Christmas and the New Year). Operators will be encouraged to make full use of these facilities so that special timetables are available for public information systems well in advance of each special day of operation, and to avoid the need to submit special registrations (or notifications) for such services.
The TransXChange Publisher is a free tool issued along with TransXChange, which allows users to render TransXChange XML documents into a readable timetable-like layout, that uses the Acrobat pdf file format. See Figure 9-1. The free Acrobat reader from Adobe Inc. (http://www.adobe.com) can be used to read and print pdf files.

The Publisher can be invoked from a Desktop GUI. It has options to produce:

- **Particulars.** The particulars section includes a summary of the contents of the TransXChange document, (for example how many stops and journeys) followed by a textual listing of the entities described in the file (such as operators, services, routes, and stops).

- **Timetable.** The timetable section contains matrix timetables for the services in the TransXChange document. Separate timetables are generated for different services, directions (e.g. outbound and inbound), and day types (e.g. a Monday to Friday timetable, and a Saturday timetable).

- **Diagnostic Report.** The diagnostics section contains a report detailing violations of consistency checks for the TransXChange document (over and above those expressed in the TransXChange XML Schemas alone).

- **Route Track.** The route track section is a separate pdf document. It consists of route plots for the services in the TransXChange on a map background along with an accompanying table of stops. It requires an on-lin connection to use.

**Figure 9-1 – Publisher**

### 9.1 Required Environment

The TransXChange Publisher requires the installation of a standard open source environment for running Java (Java Runtime Environment 1.4.2 or higher). See Installation instructions for platform requirements.
The Route Track option requires a broadband internet connection to access the web services that provide stop and map data.

9.2 Installation Process

The Publisher is available as a downloadable zip at http://www.transxchange.org.uk.

Installation instructions and examples are included on the site.

9.3 Run Time Options

The Publisher has a number of run time options
(a) To control the content to be included.
(b) To specify various aspects of the rendering of content.

9.4 Generalised list of Publisher parameters

<table>
<thead>
<tr>
<th>Group</th>
<th>Parameter</th>
<th>Data type</th>
<th>Default</th>
<th>Description</th>
<th>WS</th>
<th>Comm and Line</th>
<th>GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Operands</td>
<td>Document-Path</td>
<td>url</td>
<td>Required</td>
<td>Name and Path to TransXChange XML document and associated files that are</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>to be published.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Operands</td>
<td>OutputPath</td>
<td>url</td>
<td>Optional</td>
<td>Output directory in which to place published output. If not otherwise</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>specified, output is placed in the same directory as the input document.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td>ValidateXML</td>
<td>boolean</td>
<td>true</td>
<td>Apply XML validation</td>
<td>Y</td>
<td>Y+</td>
<td>Y</td>
</tr>
<tr>
<td>options</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output Section</td>
<td>Auto</td>
<td>Vosa</td>
<td>Options</td>
<td>Options controlling the interpretation of auto</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Content Options</td>
<td></td>
<td>/vosaAll /full</td>
<td></td>
<td>See Parameter defaults below -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Particulars</td>
<td>none / basic / full</td>
<td>full</td>
<td>Include the particulars in output.</td>
<td>Y</td>
<td>Y [1]</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• auto – default by pub format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• none – no particulars</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• basic – basic particulars</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Timetable</td>
<td>none / basic / full</td>
<td>full</td>
<td>Include the timetable matrix in output.</td>
<td>Y</td>
<td>Y [1]</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• auto – default by pub format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• none – no matrix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• basic – Omit footnotes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• full - Include the timetable footnotes in output.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RouteTrack</td>
<td>none / plain / basic / tiled</td>
<td>none</td>
<td>Include the route track in output. Default is false.</td>
<td>Y</td>
<td>Y+</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• auto – default by pub format</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• none – no route track</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• plain – no map tiles</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• basic – Omit stop list</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 8-9 – Publisher Interface Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embed</td>
<td>boolean</td>
<td>true</td>
<td>Include any embedded image content in output</td>
</tr>
<tr>
<td>Diagnostics</td>
<td>auto</td>
<td>auto</td>
<td>Publish a diagnostic section.</td>
</tr>
<tr>
<td>Filters</td>
<td>all</td>
<td>All [PTP]</td>
<td>Include timing points of this type.</td>
</tr>
<tr>
<td>MergeFrequency</td>
<td>boolean</td>
<td>true</td>
<td>Merge similar frequent journeys into a single column.</td>
</tr>
<tr>
<td>RouteMap</td>
<td>Auto</td>
<td>Auto</td>
<td>Scale to use when tiling map. Small: 1:10,000, Large 1:500:00</td>
</tr>
<tr>
<td>RouteGrouping</td>
<td>Single</td>
<td>false</td>
<td>One route per map, or per direction.</td>
</tr>
<tr>
<td>RouteTiling</td>
<td>A4</td>
<td>A4</td>
<td>Output as A4 tiles or single image.</td>
</tr>
<tr>
<td>StopData</td>
<td>local</td>
<td>Web-Service</td>
<td>Source of stop coordinates.</td>
</tr>
<tr>
<td>MapData</td>
<td>none</td>
<td>Web-Service</td>
<td>Source of map tiles. Only used if RouteTrackMap specified.</td>
</tr>
<tr>
<td>Watermark</td>
<td>Vosa</td>
<td>Controls image.</td>
<td></td>
</tr>
<tr>
<td>Rendering</td>
<td>Output-Format</td>
<td>pdf</td>
<td>Output format pdf</td>
</tr>
</tbody>
</table>

[1] Command line by suppressing other parts: timetableOnly.
[3] full
[4] Matrix only/ HTML output is a Debug Tool -

#### 9.5 Publishing Actions

The Publisher publishes a document in the following order:

9.5.1 Particulars & Service Matrix

- Summary Page
- Operator
- Serviced Organisations
- Services
- Registrations
  - ShortNoticeRegistrations
- Lines
- Routes
  - Local Stop Declarations
  - References to existing stops
  - Embedded Map
- Fixed Route Services
  - Outbound VehicleJourneys
    - Monday to Friday Built-in Journey Grouping
      - Matrix
      - Notes
    - Saturday Built-in Journey Grouping
      - Matrix
3. Sunday Built-in Journey Grouping
   a. Matrix
   b. Notes

4. Custom Journey Groupings
   a. Matrix
   b. Notes

ii. Inbound Vehicle Journeys
   1. Monday to Friday Built-in Journey Grouping
      a. Matrix
      b. Notes
   2. Saturday Built-in Journey Grouping
      a. Matrix
      b. Notes
   3. Sunday Built-in Journey Grouping
      a. Matrix
      b. Notes
   4. Custom Journey Groupings
      a. Matrix
      b. Notes

- Flexible Route Services
  i. Flexible Stops
  ii. Fixed Stops
  iii. Timebands

- Supporting documents

9.5.2 Route Map

- Outbound
  i. Service Map
  ii. List of Outbound route Stops

- Inbound
  i. Service Map
  ii. List of inbound Route Stops
10 NAMING & CODING CONVENTIONS

Systematic naming conventions and a consistent coding style are used in the TransXChange schemas. These conventions are summarised in this section.

10.1 Naming of Elements

TransXChange follows consistent principle for naming elements:

10.1.1 Use of Camel Case

Camel case is used for all names in the XML schema:

- Upper camel case is used for element and attribute names, for example JourneyPatternTimingLink, HailAndRide.
- Lower case is however used for two standard attributes: xsd:lang and id, following W3C usage.
- Lower camel case is preferred or enumerated character values, for example ‘saturdayMorning’, except for proper names, which may be capitalised, e.g. ‘IsleOfMan’
- Acronyms are treated as words for capitalisation, thus TanCode, not TANCode. This is one point where we follow common best practice but diverge from e-gif. Treating acronyms as words allows for a uniform parsing of names to derive their components, and avoids ambiguity on case of contiguous acronyms, for example TANAPD vs. TanApd, or one letter words contiguous with an acronym, for example DialATAN vs. DialATan.

10.1.2 Use of Standard Name Suffixes

TransXChange and NaPT schema element, type and attribute names have been revised along consistent principles:

- All simple types end with the suffix ‘Type’.
- All complex types end with ‘Structure’.
- All enumerations end with ‘Enumeration’.
- All groups end with ‘Group’.
- Elements representing references to other entities are suffixed with ‘Ref’.
- Externally referenced identifiers of entities are generally suffixed with ‘Code’ (and represented as elements). Code values are usually unique for the element type within a document.
- Internally referenced identifiers are generally named with ‘id’ (and represented as attributes). id attributes typically have a keyref constraint on their uniqueness. The uniqueness scope for id attributes is normally for the element type within an instance document, but could also be just within an instance of specified element.
- Externally referenced classifiers of entities are generally suffixed with ‘Classification’ (rather than say ‘Type’). (Some exceptions are made to this rule for legacy usage).
- Externally referenced names of entities are generally suffixed with ‘Name’. If the context is readily apparent they may be called just Name.
- Natural Language text descriptions of entities are generally termed ‘Description’.

10.1.3 Meaningful Names

Several other consistent naming principles are followed:

- Abbreviations are generally avoided – for example ‘Operations’ is preferred to ‘Op’.
- A container element representing a one-to-many relationship is in the plural; for example, StopPoints contains one or more StopPoint elements.
- We avoid repeating the name of the parent element as an adjective in individual child elements, except where for semantically important elements. Thus for example, Author contains Title, Position, Forename, Surname, not AuthorTitle, AuthorPosition, AuthorName, AuthorSurname
- We avoid the use in domain elements names of terms that have strong software connotations:
o The suffixes ‘Type’ and ‘Group’ are avoided in element names except for internal schema elements.
o The term ‘Exclusion’ is used generically to denote an exclusion period for the service (rather than the previous term Exception) e.g. JourneyPatternExclusion.

10.1.4 Semantically Significant Order

Several principles are used to order the subelements at any given level of containment:
- When declaring elements within a parent, subelements are placed in a consistent general order according to the nature of their role as follows:
  (i) Elements that identify the entity, such as codes or numbers.
  (ii) Elements that classify the entity.
  (iii) Elements that describe the element in text, such as names or descriptions.
  (iv) Elements describing other properties of the entity.
- Where there is an inherent temporal order, elements are placed in temporal sequence.

10.1.5 Standardised Terminology

An attempt has been made to use the appropriate Transmodel term wherever appropriate. For example Garage rather than Depot. The main divergences from Transmodel are listed in section 13.2.

10.2 Typing of Elements

Some general principles are used for typing values.
- Explicit, specific types are used wherever possible, for example Duration:
- Complex types are declared for all significant elements.
- Internally referenced identifiers are generally of type IdType, which is defined to be of base XML type NMTOKEN.
- Elements whose content is a text string in a national language are of type NaturalLanguageStringStructure.

10.3 Element Constraints

Some general principles are used for constraining values.
- Mandatory Elements are normally populated. XML constraints are usually specified to ensure mandatory elements are populated, for example strings should contain at least one character.
- Optional elements not empty: Where alternative structures are available, the absence of an element is not relied upon to infer meaning. Instead an empty element or attribute value is used to make the condition explicit, or there is a default value defined. This principle has been generally been followed for new and remodelled features.

10.4 Use of Attributes

In TransXChange, XML element attributes are generally used only for metadata, that is, data about data, such as change dates, or internal identifiers. Table 10-1 summarises the attributes used in TransXChange.

<table>
<thead>
<tr>
<th>Group</th>
<th>Element</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>TransXChange root element.</td>
<td>CreationDateTime</td>
</tr>
<tr>
<td>Version</td>
<td></td>
<td>ModificationTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FileName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SchemaVersion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modification,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RevisionNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DataSource</td>
</tr>
<tr>
<td>Entity</td>
<td>StopPoint, StopArea, NptgLocality, Service,</td>
<td>CreationDateTime</td>
</tr>
<tr>
<td>Version</td>
<td>VehicleJourney, FlexibleVehicleJourney, Route,</td>
<td>ModificationTime</td>
</tr>
<tr>
<td></td>
<td>RouteLink, FlexibleZone, Registration, JourneyPattern,</td>
<td>Modification,</td>
</tr>
<tr>
<td></td>
<td>Operator, JourneyPatternInterchange,</td>
<td>RevisionNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 10.5 Implementation of Model Relationships

In **TransXChange**, some stylistic conventions are used to make clear the mapping of the reference model relationships into the XML schema.

- Significant entities have a uniquely scoped identifier (always an element named *xxxCode*, or *xxxNumber*, or an id attribute).
- Relationships are implemented by placing a reference to the identifier as a foreign key on the referencing element (shown by the navigability arrow in UML diagrams). The reference has the form *xxxRef*. For example, **StopPoint** is identified by an *AtcoCode*, and referenced in relationships by a *StopPointRef*.
- Container elements are generally used for significant one-to-many relationships, for example **StopPoints** contains the **StopPoint** elements. In the Delta schema can be marked as ‘delta’ to indicate that only a subset of elements are supported.

---

**Table 10-1 – TransXChange Attributes**

<table>
<thead>
<tr>
<th>Child Entity Version</th>
<th>OperatorLicence, Descriptor, FlexibleServiceTimes, AnnotatedCrossRef, StopValidity, VariableStopAllocation</th>
<th>As above 2.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Route</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>JourneyPattern</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>DeadRun</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>RouteSection</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternSection</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>RouteLink</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternTimingLink</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>VehicleJourneyTimingLink</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>PositioningLink</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternStopUsage</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>Vehicle.JourneyTimingLink</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternInterchange</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>VehicleJourneyInterchange</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>JourneyGrouping (Custom &amp; Built-in)</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>ConnectingVehicleJourney</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>Calendar</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>OperatingDay</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>DayAssignment</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>1.2</td>
</tr>
<tr>
<td>Data</td>
<td>Location</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>JourneyPatternStopUsage, VehicleJourney</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Route / Track / MapSystemReference</td>
<td>2.0</td>
</tr>
<tr>
<td>Language</td>
<td>Text elements: Name, Description, etc. See section on National Language Support</td>
<td>2.0</td>
</tr>
</tbody>
</table>
11 NATIONAL LANGUAGE SUPPORT

TransXChange is enabled to allow the coding of schemas in different National Languages, such as Welsh.

11.1 Text Content Types

The textual data of a TransXChange schedule falls into three different categories:

- **Fixed Text**: National Language Translations of fixed encoded TransXChange values (for example the TAN area names), and terminology for concepts such as ‘Service’ rendered when using a style sheet to transform a schedule into a published format.

- **Free Text**: The contents of data elements that can be specified as content for textual elements (having an xml:lang attribute and a type of NaturalLanguageStringType), for example operator names, route descriptions and other notes.

- **External Data**: The contents of data fetched from external data systems, for example NaPTAN stop names.

11.1.1 Use of Fixed Text

An overall xml:lang attribute is specified at the schema level on the TransXChange root element. This specifies the default language for the schedule, i.e. the default implied language that is to be used to publish the timetable. It defaults to English.

- Translations can be established for the text associated with the different fixed elements.

11.1.2 Use of Free Text

Elements which may contain free text in a natural language (Table 11-1), such as Welsh or English, have an xml:lang language attribute to indicate the language in which they are.

- English is assumed if no attribute is specified.

- The provision of alternative names for a stop in different languages is covered by NaPTAN, which allows for multiple alternative names.

<table>
<thead>
<tr>
<th>Group</th>
<th>Element</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>StopPoint</td>
<td>CommonName</td>
<td>Use</td>
</tr>
<tr>
<td></td>
<td>NptgLocalityName</td>
<td>NaPTAN</td>
</tr>
<tr>
<td></td>
<td>StopPoint / Short CommonName</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Landmark</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Street</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Crossing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Indicator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Town</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Suburb</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopPoint / Note</td>
<td></td>
</tr>
<tr>
<td>StopArea</td>
<td>StopArea / Name</td>
<td></td>
</tr>
<tr>
<td>Organisation</td>
<td>ServicedOrganisation / Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DatePattern / Description</td>
<td></td>
</tr>
<tr>
<td>FlexibleZone</td>
<td>FlexibleZone / Description</td>
<td></td>
</tr>
<tr>
<td>Route</td>
<td>Route / Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Route / Maneuvre</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Instruction / Summary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Track / Feature / OnwardName</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Track / Feature / Description</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>Origin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Destination</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vias /ViaName</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ServiceClassification / OtherService / Description</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>StopRequirements / NewStopsRequired / Note</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note / NoteText</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ToBeMarketedWith / RelatedService / Description</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td>ContractedService / Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QualityPartnership</td>
<td></td>
</tr>
</tbody>
</table>
11.1.3 External Data

Any national language alternatives of StopPoint and StopArea names are provided by NaPTAN. The schema xsd:lang attribute should be used to determine the preferred language alternative to use when rendering names in timetables.

11.2 Publishing or Exchanging Documents

Note that the free text elements may only be in one language at a time in a given document. In order for the language specific free text elements of a schedule to be exchanged in multiple languages, the schedule must be republished in each language in turn.
12 VERSIONING

TransXChange schemas and documents must be versioned with an explicit version number so as to manage change in a distributed operating environment, and in particular to allow the inter-operability of versions of TransXChange running concurrently on different systems.

12.1 Version Numbering Convention

TransXChange follows the e-Gif convention for version numbering.

- Released Version numbers have the form n.m, (e.g. ‘3.0’).
- Drafts have the form n.mx (e.g. ‘3.1a’).
- The main version number (n) will be incremented when the change from the previous version of the schema will cause existing documents to fail to validate. For example if a new mandatory element is added.
- The minor version number (m) will be incremented when the change to the schema will allow existing documents to continue to validate. However some new documents may fail to validate against the old version (for example, if a new optional element is added).
- The draft version number (x) indicates that the version is still under discussion and may be subject to further changes. Generally it will be incremented to indicate a material change to a previous release or previous draft. Intermediate drafts will usually be withdrawn once they are superseded.

12.2 Resource Versions

12.2.1 Schema URI Version

In line with W3C practice, a separate directory and URL will be used for each version of the schemas; the schema name will remain the same (N.B. a directory rather than document level numbering system is preferred for the leaf schemas because it facilitates the management of multiple components of a modularised schema, and multiple document artefacts).

For example

- http://www.transxchange.org.uk/schemas/2.0f/TransXChange_registration.xsd

and

- http://www.transxchange.org.uk/schemas/3.1/TransXChange_general.xsd

Different versions will coexist at the same time. Old versions will generally first be deprecated, and then retired.

12.2.2 Namespace URI Version

e-GIF mandates that Namespace URI should not be versioned. (A different URL for the namespace and the schema) The following URI will be used for namespace.

- http://www.transxchange.org.uk/schemas/

12.2.3 Package Versions

TransXChange embeds a number of common type definition packages that are shared with other UK standards. For convenience, a separate copy of the common packages is distributed with each standard. The individual package files are given version numbers in line with the e-GIF system in order to ensure the correct version is used.

For example, for the shared NaPT stop definition types file might be called NaPT_stop-v1-0.xsd. It will be distributed in TransXChange as:

- http://www.transxchange.org.uk/schemas/2.0/napt/NaPT_stop-v1-0.xsd
12.3 Packages

The TransXChange model is modularised into a number of packages, with a strict linear dependency. See Figure 12-1 & Figure 12-2.

Figure 12-1 – TransXChange Packages
The XML schemas are organised according to corresponding package groups (see Table 12-1). *TransXChange* schemas are placed in the root folder, prerequisite shared schemas are placed in subfolders (\apd, \napt and \xml).

<table>
<thead>
<tr>
<th>Standard</th>
<th>Folder</th>
<th>Schemas</th>
<th>Contents</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>TransXChange</em></td>
<td>root</td>
<td><em>TransXChange_registration.xsd</em></td>
<td>Terminal schema for Registrations.</td>
<td>Renamed in 2.0.</td>
</tr>
<tr>
<td></td>
<td>root</td>
<td><em>TransXChange_general.xsd</em></td>
<td>Terminal schema for General use.</td>
<td>Renamed in 2.0.</td>
</tr>
<tr>
<td></td>
<td>root</td>
<td><em>TransXChange_general_delta.xsd</em></td>
<td>Terminal schema for Delta use.</td>
<td>New in 2.4.</td>
</tr>
<tr>
<td>TXC</td>
<td>\xc</td>
<td>TXC_authorities-v2-3.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_common-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_connectingVehicleJourney-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_deadRun-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_flexibleJourneyPattern-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_frequency-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_garage-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journey_support-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeyPattern_support-v2-3.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeyPatternInterchange-v2-3.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeySection-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeyPatternTimingLink-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeyPatternTimingLink-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_journeyPatternTimingLink-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_operatingDay-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_operational-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_operator-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_passingTimes-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_registration-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_route-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_serviceJourneyGrouping-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_service_parts-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_serviceCalendar-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_serviceOrganisation-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_service-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_utility_times-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_utility_types-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_validity-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleEquipment-v2-55.xsd</td>
<td></td>
<td>New in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleJourney-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleJourney_support-v2-3.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleJourneyInterchange-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleJourney-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\xc</td>
<td>TXC_vehicleType-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.5.</td>
</tr>
<tr>
<td>NaPT</td>
<td>\napt</td>
<td>NaPT_all.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_dates-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_dayTypes-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_location-v2-5.xsd</td>
<td></td>
<td>Renamed in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_operatorSupport-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_modes-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_pointOfInterest-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_stop-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_stopAccessibility-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_stopArea-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_tariffZone-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_utility_rights-v2-3.xsd</td>
<td></td>
<td>New in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_utility_types-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_utility_units-v2-4.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_utility_xml-v2-1.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_versioningAttributes-v2-2.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
</tr>
<tr>
<td>NaPT_netex</td>
<td>\napt</td>
<td>NaPT_netex_accessibility-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_netex_entity_support-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td></td>
<td>\napt</td>
<td>NaPT_netex_equipment-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
</tr>
<tr>
<td>Scheme</td>
<td>Module Name</td>
<td>Version</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>NAPT</td>
<td>NaPT_netex_equipmentVehiclePassenger-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
<td></td>
</tr>
<tr>
<td>NAPT</td>
<td>NaPT_netex_facility_support-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
<td></td>
</tr>
<tr>
<td>NAPT</td>
<td>NaPT_netex_facility_version-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
<td></td>
</tr>
<tr>
<td>NAPT</td>
<td>NaPT_netex_localService-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
<td></td>
</tr>
<tr>
<td>NAPT</td>
<td>NaPT_netex_travelRights_types-v2-5.xsd</td>
<td></td>
<td>New in 2.5.</td>
<td></td>
</tr>
<tr>
<td>NPTG</td>
<td>NPTG_locality-v2-4.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
<td></td>
</tr>
<tr>
<td>NPTG</td>
<td>NPTG_locality_support-v2-4.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
<td></td>
</tr>
<tr>
<td>NPTG</td>
<td>NPTG_administrative-v2-5.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
<td></td>
</tr>
<tr>
<td>NPTG</td>
<td>NPTG_administrative_support-v2-4.xsd</td>
<td></td>
<td>Modularised in 2.4.</td>
<td></td>
</tr>
<tr>
<td>GovTalk</td>
<td>add</td>
<td>AddessTypes.xsd</td>
<td>UK address types</td>
<td>Referenced in 2.0</td>
</tr>
<tr>
<td>GovTalk</td>
<td>apd</td>
<td>CommonSimpleTypes.xsd</td>
<td>UK simple types</td>
<td>Referenced in 2.0</td>
</tr>
<tr>
<td>W3C</td>
<td>xml</td>
<td>XML.xsd</td>
<td>Standard definitions of types</td>
<td>Referenced in 2.0</td>
</tr>
</tbody>
</table>

**Table 12-1 – TransXChange 2.4 Module Names**
12.4 Version Identifiers & Change Tracking

12.4.1 Schema Version Identifier

The TransXChange schema has an explicit version attribute on it, as recommended by e-GIF.
- The schema id is “TransXChange”.
- The version identifier follows the versioning scheme e.g. “3.0”.

12.4.2 Indicating Versions on Data

In each XML instance document conforming to TransXChange, the root TransXChange element has an attribute that is populated to indicate the schema version, as recommended by e-GIF. This allows any application which processes the document to decide how to handle the document. See Table 12-2. The Schema version is one of a standard set of Content change attributes that are specified on the route elements of all NaPT schemas.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationDateTime</td>
<td>Date and Time stamp, ISO format</td>
<td>When document was first created</td>
</tr>
<tr>
<td>ModificationDateTime</td>
<td>Date and Time stamp, ISO format</td>
<td>When document was last updated, e.g. to change a workflow attribute.</td>
</tr>
<tr>
<td>Modification</td>
<td>R Nature of modification: one of new, delete, revise, delta</td>
<td>Nature of changes in document. Normally revise (New and changes)</td>
</tr>
<tr>
<td>RevisionNumber</td>
<td>Monotonically incrementing number</td>
<td>Sequentially incrementing number. May be populated by systems that track individual export sessions.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>Schema Version number</td>
<td>Version number of document</td>
</tr>
<tr>
<td>ChangesSince</td>
<td>Date and Time stamp, ISO format</td>
<td>For deltas only.</td>
</tr>
</tbody>
</table>

Table 12-2 – TransXChange Document Version Attributes

12.4.3 Data Element Version

Most significant entities in TransXChange have an optional set of a standard change attributes on them, including a modification date and revision number that can be used to specify their version level. See Table 12-3. These can be used by those wishing to support fine-grained round trip version management of individual entities. Fine grained versioning is not required for EBSR.

<table>
<thead>
<tr>
<th>Change Attributes</th>
<th>Value Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreationDateTime</td>
<td>Date and Time stamp, ISO format</td>
<td>Timestamp at creation of entity. Should be set when the entity is first created, and not subsequently be changed.</td>
</tr>
<tr>
<td>ModificationDateTime</td>
<td>Date and Time stamp, ISO format</td>
<td>Timestamp at most recent update. Should be changed every time an entity is changed, or when any of its child elements 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.</td>
</tr>
</tbody>
</table>
| Modification      | Nature of modification: one of new, delete, revise, delta | 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 against 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. |
The RevisionNumber. The revision number of 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 originator of the data 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.

Table 12-3 – Entity Change Tracking Attributes

- Timestamps should be in standard ISO format, for example '2004-04-14T14:20:00-05:00'
- The RevisionNumber of an element should be incremented (and its Modification value be set to of 'revised'), if any of its element values, attribute values or contained values is modified. It may be set to zero for a new entity.

Figure 12-3 shows the common TransXChange versioning attributes

```class Versioning Model
«interface»
Versionable
CreationDateTime  :dateTime
ModificationDateTime  :dateTime
Modification  :ModificationEnum
RevisionNumber  :RevisionNumberType
Status  :StatusEnum
BaselineVersion  :RevisionNumberType

«enumeration»
ModificationEnum
new
delete
revise
archive
delta

<enumeration>
StatusEnum
active
inactive
pending

Name:   Versioning Model
Author: nick
Version: 1.0
Created: 17/09/2009 15:52:33
Updated: 20/05/2013 15:10:41```

Figure 12-3 – UML diagram of Version Attributes
12.4.4 Change Trackable Entities

The TransXChange entities which can be change tracked are shown in Table 12-4.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Versioning</th>
<th>TXC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransXChange</td>
<td>SchemaVersion + Document Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>StopPoint</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>StopArea</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>NptgLocality</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>FlexibleZone</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Route</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>RouteSection</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>RouteLink</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Track</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>JourneyPattern</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternSection</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternStopUsage</td>
<td>No – within JourneyPatternSection.</td>
<td>2.4</td>
</tr>
<tr>
<td>JourneyPatternTimingLink</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyPatternInterchange</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Operator</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>LicencedOperator</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Garage</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>Service</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>JourneyGrouping</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>Registration</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Line</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>VehicleJourney</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>ConnectingVehicleJourney</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>FlexibleVehicleJourney</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>VehicleJourneyTimingLink</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>VehicleJourneyStopUsage</td>
<td>No – No within VehicleJourney.</td>
<td>2.4</td>
</tr>
<tr>
<td>VehicleJourneyInterchange</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>DeadRun</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>PositioningLink</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>LayoverPoint</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>ServicedOrganisation</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>ServiceFacilitySet</td>
<td>Change Attributes</td>
<td>2.5</td>
</tr>
<tr>
<td>VehicleEquipment</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>DayType</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>SupportingDocument</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>Calendar</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>StopValidity</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>OperatorLicence</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>Descriptor</td>
<td>Change Attributes</td>
<td>2.1</td>
</tr>
<tr>
<td>FlexibleServiceTimes</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>AnnotatedCrossRef</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
<tr>
<td>VariableStopAllocation</td>
<td>Change Attributes</td>
<td>2.4</td>
</tr>
</tbody>
</table>

Table 12-4 – TransXChange Tracked Data Elements

12.5 Exchange of Deltas

EBSR requires the submission of complete coherent schedules using the TransXChange Registration Schema. The schema includes integrity constraints to check that referenced elements are present. The TransXChange General Schema may be used to exchange coherent subsets of a Schedule, as well as lists of stops, serviced Organisations, etc. It likewise includes some referential integrity checks.

Other applications may wish to exchange just the changes to a particular schedule or set of elements. This may be done using the TransXChange Delta schema. This is a special version of the TransXChange schema that has no referential integrity constraints. It is possible to mark whether the set of elements of a specific type is partial or complete using the modification attribute, which may take the values ‘delta’ (changes only) ‘revise’ (all current & new instances), or ‘new’ (new instances only. The delta.xml example provides an example of a delta exchange.
12.6 Names of TransXChange Files

When dealing with a large number of bus schedules, it is helpful for document management if the file name used for a bus schedule when it is exchanged as an XML document gives an indication of its contents. The following format is recommended for file names of TransXChange XML documents:

```
Line_Operator_Area_ServiceCode_StartDate.xml
```

Where:

- **Line** is the service number seen by the public, as defined by a `Service / Lines/ Line` element within the document. If there is more than one `Line` associated with a service, use the first.
- **Operator** identifies the service operator and is either:
  - The operator code, i.e. `RegisteredOperator / OperatorCode` element for the service specified within the document.
  - The operator license number, i.e. `Operator / LicenceNumber` of the operator registering the service.
- **Area** identifies the service area and is either:
  - `Area Code`: Three digit ATCO database code for the district/authority 450. This is the `NPTG AdministrativeArea / AtcoAreaCode`.
  - `TAN Code`: Two character TAN prefix. This is the `Registration / VosaRegistrationNumber / TanCode` specified within the document.
- **ServiceCode** is an arbitrary unique identifier for the service as specified by a `Service / ServiceCode` element within the document.
- **StartDate**: Is the registered start date of the service as defined by a `Service / OperatingPeriod / StartDate` within the document.

So for example, the 757 service operated by Aztecbird (AZT) in West Yorks (450), the general TransXChange export file name would be:

Using the operator code:

```
757_AZT_450_4431_20020428.xml
```

Using the operator licence number:

```
757_3888_450_4431_20020428.xml
```

Using the Tan prefix on a registration:

```
757_3888_PB_4431_20020428.xml
```

For registrations there should generally be a separate file for each registration change date, i.e. one file for the initial service, one for a new version of the service starting 01/07/2004 and so on.

When exchanging between the authority databases and journey planner and real time systems, multiple services may be contained in a single file, using the general schema. In this case there is no preferred naming scheming.
13 TRANSMODEL & TRANSXCHANGE COMPARISON

13.1 Transmodel Principles

TransXChange is based on Transmodel, a general abstract model for describing public transport information systems, devised on carefully elaborated informational science principles. Some of the key principles for Transmodel may be summarised as follows:

1. **Layered Semantic Models:** The efficient modelling of public transport information requires a number of distinct models, representing different levels of discourse. For example, (i) the geospatial location (i.e. map) layer, (ii) the network topology layer, (iii) the service pattern layer, (iv) the timed vehicle journey layer, (v) the operational running layer, etc.

2. **Projection:** It should be possible to combine the different models in order to compute over them, relating the corresponding elements of different levels of discourse precisely and unambiguously, using a common frame of reference. For example, route links should map onto geospatial objects such as roads; timing links should map onto route links, etc. The establishment of equivalences between distinct model layers is termed **projection**.

3. **Common Terminology:** A standard set of common conceptual entities should be used for the elements making up the models at each different layer, and a standard Transmodel terminology should be used. For example, Line, Journey Pattern, Vehicle Journey, Location.

4. **Point and Link Structures:** Public Transport Information System models typically involve complex networks which are modelled in computer systems by graphs; that is, as networks of nodes (points) and edges (links). Depending on the information of interest in a particular application, it may be appropriate to use ordered collections of links, ordered collections of points, or combinations thereof. Links of a given type should only connect to points of the corresponding semantic level of discourse. Only one unambiguous sequence of points (whether modelled as a point sequence, or link sequence) may be used in a given journey or service pattern.

5. **Well-defined Data Systems.** Elements corresponding to external entities should be assigned unique identifiers from agreed data reference systems.
13.2 Transmodel Terminology

Wherever possible, TransXChange follows Transmodel terminology for PT concepts. The equivalences between some key TransXChange elements and their corresponding Transmodel concepts are shown in Table 13-1. Divergences are highlighted in bold.

<table>
<thead>
<tr>
<th>Transmodel</th>
<th>TransXChange 2.0</th>
<th>Previously 1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMINISTRATIVE ZONE / AREA</td>
<td>Administrative Area</td>
<td>--</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>Activity</td>
<td>ActivityFlag</td>
</tr>
<tr>
<td>Alight</td>
<td>Set down</td>
<td>Set down</td>
</tr>
<tr>
<td>Board</td>
<td>Pick up</td>
<td>Pick up</td>
</tr>
<tr>
<td>BLOCK</td>
<td>Block</td>
<td>--</td>
</tr>
<tr>
<td>DAY TYPE</td>
<td>OperatingProfile / RegularDayType</td>
<td>DayType / GeneralOpClassification</td>
</tr>
<tr>
<td></td>
<td>PeriodicDayType</td>
<td>Periodic</td>
</tr>
<tr>
<td></td>
<td>ServicedOrganisationDayType</td>
<td>SchoolOp</td>
</tr>
<tr>
<td>DEAD RUN</td>
<td>DeadRun</td>
<td>--</td>
</tr>
<tr>
<td>DESTINATION DISPLAY</td>
<td>DestinationDisplay</td>
<td>DynamicDestinationDisplay</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>Distance</td>
<td>Distance</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>Direction</td>
<td>JourneyDirection</td>
</tr>
<tr>
<td>FARE STAGE</td>
<td>FareStage</td>
<td>--</td>
</tr>
<tr>
<td>FARE ZONE</td>
<td>FareZone</td>
<td>--</td>
</tr>
<tr>
<td>JOURNEY PATTERN</td>
<td>JourneyPattern</td>
<td>JourneyPattern</td>
</tr>
<tr>
<td>JOURNEY PATTERN LINK IN SEQUENCE + TIMING LINK</td>
<td>JourneyPatternTimingLink</td>
<td>JourneyPatternTimingLink</td>
</tr>
<tr>
<td>JOURNEY PATTERN LAYOVER</td>
<td>JourneyPattern / Layover</td>
<td>DefaultRunTime</td>
</tr>
<tr>
<td>JOURNEY PATTERN RUN TIME</td>
<td>JourneyPattern / RunTime</td>
<td>DefaultRunTime</td>
</tr>
<tr>
<td>JOURNEY PATTERN WAIT TIME</td>
<td>JourneyPattern / WaitTime</td>
<td>DefaultWaitTime</td>
</tr>
<tr>
<td>LINE</td>
<td>Line</td>
<td>(ServiceId)</td>
</tr>
<tr>
<td>(FARE SECTION ) (LINK SEQUENCE)</td>
<td>RouteSection</td>
<td></td>
</tr>
<tr>
<td>LOCATION</td>
<td>Location</td>
<td>Geocode</td>
</tr>
<tr>
<td>LOCATING SYSTEM</td>
<td>LocatingSystem</td>
<td>(Geodata system)</td>
</tr>
<tr>
<td>OPERATOR</td>
<td>Operator</td>
<td>Operator</td>
</tr>
<tr>
<td>PLACE</td>
<td>Place</td>
<td>Locality</td>
</tr>
<tr>
<td>ROUTE</td>
<td>Track (See 13.3.1)</td>
<td>--</td>
</tr>
<tr>
<td>ROUTE LINK</td>
<td>TrackLink (See 13.3.1)</td>
<td>--</td>
</tr>
<tr>
<td>RUN TIME</td>
<td>Run time</td>
<td>Run time</td>
</tr>
<tr>
<td>SERVICE</td>
<td>Service, StandardService</td>
<td>OverallServiceDescription</td>
</tr>
<tr>
<td></td>
<td>FlexibleService</td>
<td></td>
</tr>
<tr>
<td>SERVICE PATTERN</td>
<td>Route (See 13.3.1)</td>
<td>Route</td>
</tr>
<tr>
<td>SERVICE JOURNEY PATTERN LINK</td>
<td>RouteLink (See 13.3.1)</td>
<td>RouteLink</td>
</tr>
<tr>
<td>SERVICE JOURNEY PATTERN INTERCHANGE</td>
<td>JourneyPatternInterchange</td>
<td>JourneyPatternInterchange</td>
</tr>
<tr>
<td>SERVICE JOURNEY INTERCHANGE</td>
<td>VehicleJourneyInterchange</td>
<td>VehicleJourneyInterchange</td>
</tr>
<tr>
<td>SITE</td>
<td>Landmark</td>
<td>Landmark</td>
</tr>
<tr>
<td>STOP POINT</td>
<td>StopPoint</td>
<td>Stop</td>
</tr>
<tr>
<td>STOP AREA</td>
<td>StopArea</td>
<td>StopCluster</td>
</tr>
<tr>
<td>TIMING LINK</td>
<td>(JourneyPattern) TimingLink</td>
<td>TimingLink</td>
</tr>
<tr>
<td>TIME DEMAND</td>
<td>TimeDemand</td>
<td>--</td>
</tr>
<tr>
<td>VALIDITY PERIOD</td>
<td>ValidityPeriod</td>
<td>ValidityPeriod</td>
</tr>
<tr>
<td>VEHICLE JOURNEY</td>
<td>VehicleJourney</td>
<td>VehicleJourney</td>
</tr>
<tr>
<td>(VEHICLE JOURNEY + JOURNEY PATTERN LINK IN SEQUENCE + TIMING LINK)</td>
<td>VehicleJourneyTimingLink</td>
<td>VehicleJourneyTimingLink</td>
</tr>
<tr>
<td>VEHICLE JOURNEY RUN TIME</td>
<td>VehicleJourneyTimingLink / RunTime</td>
<td>RunTime</td>
</tr>
<tr>
<td>VEHICLE JOURNEY WAIT TIME</td>
<td>VehicleJourneyTimingLink / WaitTime</td>
<td>WaitTime</td>
</tr>
<tr>
<td>VEHICLE TYPE</td>
<td>VehicleType</td>
<td>VehicleType</td>
</tr>
<tr>
<td>VERSION</td>
<td>(RevisionNumber)</td>
<td>(RevisionNumber)</td>
</tr>
<tr>
<td>WAIT TIME</td>
<td>Wait time</td>
<td>Wait time</td>
</tr>
<tr>
<td>ZONE</td>
<td>FlexibleZone</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 13-1 – Comparison of Key Transmodel Terms
13.3 Divergences from Transmodel

Version 2.x of TransXChange converges significantly closer to Transmodel, but still contains a few significant differences in terminology that reflect TransXChange 1.2 usage, and the legacy of TransXChange’s original ATCO CIF representation. (Note too that Transmodel has also been subject to further evolution during the period of development of TransXChange). In addition TransXChange introduces additional convenience elements for implementation which can mostly be considered as views which compound other elements; for example a StopUsage groups various attributes that can be associated with either end of a TimingLink. The main outstanding differences (which may possibly be reduced in future), are as follows:

- Transmodel uses the term ROUTE to denote a physical path taken by vehicles through the network, identifying the road sections or track section being used with each stage. A Transmodel ROUTE LINK corresponds more properly to the Track (ROUTE LINK) and Mapping (POINT IN ROUTE LINK) elements of the TransXChange model.

- The TransXChange Route and RouteLink are similar to a Transmodel SERVICE PATTERN, and SERVICE LINK, that is, an abstract journey pattern, identifying a unique sequence of STOP POINTs in order that define a possible journey for a line, regardless of any actual timings.

13.3.1 TransXChange Representation of Journey Patterns

Note that TransXChange does not use what Transmodel would term a STOP IN SEQUENCE (or more specifically, STOP POINT IN JOURNEY PATTERN) representation of a journey pattern, but rather a Transmodel LINKS IN LINK SEQUENCE representation; more specifically, a sequence of journey pattern timing links (TIMING LINK IN JOURNEY PATTERN). The Transmodel abstract model allows for a separate set of SERVICE LINKs between the stop points of a service pattern or journey pattern that is distinct from the set of TIMING LINKs of the pattern, permitting multiple timings to be specified for the same route, and for some of the intermediate timing points not to be stop points (and stop points not to be timing points). Because TransXChange has historically been primarily concerned with the exchange of fully timed schedules for registration, all points in a TransXChange JOURNEY PATTERN are stop points, and TransXChange uses only timing links: the existence of a service link between two points is implied from the existence of a timing link between two stops. This simplifies the mapping of the representation to a published matrix timetable; however a consequence is that it forces a false interpolation of run times in some usages. For example, if there is a sequence of non-timing stop points in a pattern, for which there is only an overall run time, the overall run time must be arbitrarily assigned to one or more of the intermediate links in order to encode it in TransXChange.

In effect TransXChange makes a simplifying assumption that all TIMING POINTs are in effect also STOP POINTs so is able to use a combined Link abstraction that has both timing and service pattern properties. The StopUsage element

It may be appropriate to add a compatible STOP IN SEQUENCE and separate service and timing link representations to a future version of TransXChange.

13.3.2 Abbreviated Journey Patterns

In TransXChange, two practical expedients are used also to reduce the amount of data that has to be exchanged, and in particular the number of journey patterns.

- a. Short working of the underlying journey pattern is allowed in TransXChange, i.e. truncation of one or more stops of the pattern at either or both ends. Transmodel indicates that a separate journey pattern should be declared for any difference in stop sequence, which could be strictly interpreted as requiring a separate journey pattern for each short working vehicle journey variant.
b. Express journeys over a service pattern are allowed in TransXChange – i.e. provided a journey traverses a link, and goes past a stop, it may specify an activity of ‘pass’ to omit a particular stop.

In both the above cases, there is little or no informational benefit to having a separate journey pattern, and there is in any case little distinction between the above cases and the legitimate Transmodel representation of a vehicle following a ‘full’ journey pattern in real-time that for operational reasons passes stops, or terminates early.

13.3.3 Groups of Links

Another expedient TransXChange, uses to reduce the amount of data that has to be exchanged is a “link section”, that is, a reusable ordered list of Links that can be reused in one or more ROUTEs or JOURNEY PATTERN. This is particularly useful where there is corridor route with a long common section but many end variants. Link sections are an additional abstraction not found in Transmodel but can be seen as equivalent to GROUP OF LINKs being used in a specific way. Their use amounts to a requirement that there is always at least one “GROUP OF LINKS” associated with each journey pattern, but need not conflict in any way with a canonical Transmodel representation.
14 INTEGRITY RULES

14.1 Syntactic Integrity Rules

XML’s inbuilt mechanisms, including unique & keyref are used in the TransXChange schema to enforce a number of basic integrity checks of data within a TransXChange document, including enforcing uniqueness. A document must satisfy these constraints, or it is not well formed and will not be processed further by the TransXChange Publisher or other tools.

- Data types are specified for dates, times, durations and other common data types.
- Restricted values are enforced by enumerations – see individual tables of allowed values under the schema guide entry for constrained elements.
- Some additional rules for encoding formatted elements are enforced by regular expressions.
- Table 14-1 shows the other rules enforced by syntactic constraints.

<table>
<thead>
<tr>
<th>Group</th>
<th>Element / Code</th>
<th>#</th>
<th>Scope</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Scope</td>
<td>StopPoint / AtcoCode</td>
<td>C1</td>
<td>Stop codes of local StopPoint &amp; AnnotatedStopPointRef declarations must be unique within document.</td>
<td>StopPointRef instances must reference a StopPoint or AnnotatedStopPoint declaration. See also External Integrity rule N1.</td>
</tr>
<tr>
<td></td>
<td>StopArea / StopAreaCode</td>
<td>C2</td>
<td>Codes of local StopArea (Cluster) declarations must be unique within document.</td>
<td>StopAreaRef instances must reference a StopArea. See External Integrity rule N2.</td>
</tr>
<tr>
<td></td>
<td>ServicedOrganisation / ServicedOrganisationCode</td>
<td>C3</td>
<td>Codes of ServicedOrganisation declarations must be unique within operator.</td>
<td>ServicedOrganisationRef instances must reference a local definition of a ServicedOrganisation element. Any ParentServicedOrganisationRef must also be declared</td>
</tr>
<tr>
<td></td>
<td>Service / ServiceCode</td>
<td>C4</td>
<td>Code of each Service must be unique within document.</td>
<td>ServiceRef instances must refer to a local definition of a Service.</td>
</tr>
<tr>
<td></td>
<td>Garage / GarageCode</td>
<td>C6</td>
<td>Codes of Garage declarations must be unique within document.</td>
<td>GarageCodeRef instances to a Garage must reference a local definition of a Garage element.</td>
</tr>
<tr>
<td></td>
<td>OperatorCode</td>
<td>C7</td>
<td>Codes of local Operator declarations must be unique within document.</td>
<td>OperatorRef instances must refer to a local definition of a Service.</td>
</tr>
<tr>
<td>Key Uniqueness</td>
<td>StopPoint / PrivateCode</td>
<td>U1</td>
<td>PrivateCodes of StopPoint s must be unique within document</td>
<td></td>
</tr>
<tr>
<td></td>
<td>StopArea / PrivateCode</td>
<td>U2</td>
<td>PrivateCodes of StopAreas must be unique within document</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VehicleJourney / PrivateCode</td>
<td>U3</td>
<td>PrivateCodes of VehicleJourneys must be unique within document</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Route / PrivateCode</td>
<td>U4</td>
<td>PrivateCodes of Routes must be unique within document</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JourneyPattern / PrivateCode</td>
<td>U5</td>
<td>PrivateCodes of JourneyPatterns must be unique within document</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Scope</td>
<td>ID</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>------------------</td>
<td>------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ServicedOrganisation / PrivateCode</td>
<td>U6</td>
<td>PrivateCodes of ServicedOrganisations must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator / PrivateCode</td>
<td>U7</td>
<td>PrivateCodes of Operators must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service / PrivateCode</td>
<td>U8</td>
<td>PrivateCodes of Services must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service / JourneyGrouping</td>
<td>U9</td>
<td>PrivateCodes of JourneyGrouping must be unique within document (TXC v2.4).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Scope</td>
<td>ID</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Route / id</td>
<td>I1</td>
<td>id of each Route must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JourneyPattern /id</td>
<td>I2</td>
<td>id of each JourneyPattern must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line / id</td>
<td>I5</td>
<td>id of each Line must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RouteSection / id</td>
<td>I6</td>
<td>id of each RouteSection must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JourneyPatternSection / id</td>
<td>I7</td>
<td>id of each JourneyPatternSection must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RouteLink / id</td>
<td>I8</td>
<td>id of each RouteLink must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JourneyPatternTimingLink / id</td>
<td>I9</td>
<td>id of each JourneyPatternTimingLink must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VehicleJourneyTimingLink / id</td>
<td>I10</td>
<td>id of each VehicleJourneyTimingLink must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JourneyPatternStopUsage / id</td>
<td>I11</td>
<td>id of each JourneyPatternStopUsage must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VehicleJourneyStopUsage / id</td>
<td>I12</td>
<td>id of each VehicleJourneyStopUsage must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JourneyPatternInterchange</td>
<td>I13</td>
<td>id of each JourneyPatternInterchange must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VehicleJourneyInterchange</td>
<td>I14</td>
<td>id of each VehicleJourneyInterchange must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DayType / id</td>
<td>I15</td>
<td>id of each DayType must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator / id</td>
<td>I16</td>
<td>id of each Operator must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LicencedOperator / id</td>
<td>I17</td>
<td>id of each LicencedOperator must be unique within document.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DayType / id</td>
<td>I18</td>
<td>id of each DayType must be unique within document.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 14.2 Semantic Integrity Rules

Table 14-3 shows additional integrity rules that need to be applied by applications parsing a TransXChange XML document. These are subdivided into two categories:

- **Intrinsic Constraints**: Consistency checks that can be applied without reference to external data. For many of these, a sensible recovery action can be taken.
- **Extrinsic Constraints**: Checks of data values that require reference to an external source. Whether these need to be applied depends on the availability of the relevant data sets, and the purpose of the application.

Rules are assigned a severity (see Table 14-2) that indicates the likely action that an application such as TransXChange Publisher will take if the rule is not satisfied.

Rules that may affect the correct publishing of a document by the TransXChange Publisher are marked with a 'p'.

#### Table 14-1 – Syntactic Integrity Rules

<table>
<thead>
<tr>
<th>Calendar / id</th>
<th>I19</th>
<th>Id of each Calendar must be unique within document.</th>
<th>Calendar Ref instances must refer to a local definition of a Calendar.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic</td>
<td>VehicleJourneyRef</td>
<td>X1</td>
<td>VehicleJourney must not reference itself.</td>
</tr>
</tbody>
</table>

#### Table 14-2 – Severity Codes for Semantic Integrity Rules

<table>
<thead>
<tr>
<th>Group</th>
<th>#</th>
<th>Rule Name</th>
<th>Description</th>
<th>Cat</th>
<th>Sev</th>
<th>Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>Dc1</td>
<td>Valid FileName</td>
<td>File name is made up of recommended elements.</td>
<td>Int</td>
<td>6</td>
<td>Allow, but give warning.</td>
</tr>
<tr>
<td>NaPTAN</td>
<td>Na1</td>
<td>Valid NaPTAN Stop Identifiers.</td>
<td>Stop points referenced by an AnnotatedStopPointRef must exist in the NaPTAN database.</td>
<td>Ext</td>
<td>4</td>
<td>Warning.</td>
</tr>
<tr>
<td>Na2</td>
<td>Valid NaPTAN StopArea Identifiers.</td>
<td>Stop areas referenced by a StopAreaRef of a local StopPoint definition must exist in the NaPTAN database, or be defined locally.</td>
<td>Ext</td>
<td>4</td>
<td>Warning.</td>
<td></td>
</tr>
<tr>
<td>Na3</td>
<td>Local NaPTAN StopAreas.</td>
<td>Stop areas referenced by a StopAreaRef of a local StopPoint definition should belong to the same Admin Area as the StopPoint or to a national area e.g. 910.</td>
<td>Ext</td>
<td>6</td>
<td>Warning.</td>
<td></td>
</tr>
<tr>
<td>Ng3</td>
<td>Valid NPTG Localities.</td>
<td>NPTG localities referenced by NptgLocalityRef of local StopPoint definition must exist in the NPTG database.</td>
<td>Ext</td>
<td>4</td>
<td>Warning.</td>
<td></td>
</tr>
<tr>
<td>Ng4</td>
<td>Valid NPTG Administrative Areas.</td>
<td>NPTG administrative areas referenced by an AdministrativeAreaRef of local stop point definition must exist in the NPTG database.</td>
<td>Ext</td>
<td>4</td>
<td>Warning.</td>
<td></td>
</tr>
<tr>
<td>Registration</td>
<td>RG1</td>
<td>Justifications</td>
<td>Short term registrations must have at least one justification element (Severity 2 – i.e. required for submission).</td>
<td>Int</td>
<td>3</td>
<td>Warning. (+TXC 2.4)</td>
</tr>
<tr>
<td>RG2</td>
<td>New stops</td>
<td>Only Cancellation may omit new stops required</td>
<td>Int</td>
<td>4</td>
<td>Warning. (+TXC 2.4)</td>
<td></td>
</tr>
<tr>
<td>Serviced Organization</td>
<td>Eo1</td>
<td>Valid Serviced Organizations.</td>
<td>For local authorities, should be a valid DIE LEA code. For schools, should be a valid DIE school code.</td>
<td>Ext</td>
<td>5</td>
<td>Warning.</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-------------------------</td>
<td>-----</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Op2</td>
<td>Distinct Operator References.</td>
<td>Int</td>
<td>Ignore associated operator.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Op3</td>
<td>Distinct Associated Operator roles.</td>
<td>Int</td>
<td>Ignore duplicate references.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Route</td>
<td>Rs1</td>
<td>Linear routes.</td>
<td>Int</td>
<td>Reject.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rs2</td>
<td>Route section link direction.</td>
<td>Int</td>
<td>Use first direction found.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rs3</td>
<td>Route Direction Antithesis.</td>
<td>Int</td>
<td>Treat Clockwise as Outbound.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ri1</td>
<td>Route Link sequence stop references.</td>
<td>Int</td>
<td>Ignore second usage.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ri2</td>
<td>Route Link distinct endpoints.</td>
<td>Int</td>
<td>Allow, but issue warning.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RI3</td>
<td>Track end points constrained to route.</td>
<td>First and last points of Track mapping should correspond (i.e. be near to) stop points of parent RouteLink.</td>
<td>Int 3 Ignore points.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R14</td>
<td>Stop Type Usage</td>
<td>Within a given route Fixed stops (i.e. stops of type MKD) should not fall within the area of Hail and Ride stops (i.e. stops of type HAR)</td>
<td>Int 2 Report as disallowed (TXC+2.4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jp1</td>
<td>Timing endpoints.</td>
<td>Start and end stops of a journey pattern should have a StopType TimingStatus of principal point.</td>
<td>Int 4, p Treat as PTP regardless.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jp2</td>
<td>Distinct journey pattern Interchange References.</td>
<td>Inbound and outbound journey patterns at an interchange should normally be distinct.</td>
<td>Int 6 Allow, but give warning.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jp3</td>
<td>Journey pattern Direction.</td>
<td>JourneyPattern / Direction should correspond to one of the Service direction values. If Service has only a single direction value, the JourneyPattern / Direction should match. If the Service / Direction has a value of circular or inboundOrOutbound then JourneyPattern must supply an explicit override rather than using a value of inherit?</td>
<td>Int 3 Use Journey Pattern value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jps1</td>
<td>Section Projection.</td>
<td>If there are route sections, then for each JourneyPatternSection, there should be a corresponding RouteSection with the same number of links.</td>
<td>Int 1 Reject.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jps2</td>
<td>Linear journey patterns.</td>
<td>In a sequence of JourneyPatternSection instances making up a given JourneyPattern, the To / StopPoint of the last link of a given JourneyPatternSection should be the same as the From / StopPoint of the first link of the succeeding JourneyPatternSection in the JourneyPattern.</td>
<td>Int 1, p Reject.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jpt1</td>
<td>Journey Pattern timing link sequence stop references.</td>
<td>In a collection of successive timing links, 'To' stop reference of previous link should be same as 'From' stop reference of next successive link.</td>
<td>Int 6, p Ignore second usage.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jpt2</td>
<td>Journey Pattern timing link distinct endpoints.</td>
<td>'From' and 'To' stops of a timing link should be distinct, i.e. not the same</td>
<td>Int 6 Allow.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jpt3</td>
<td>Route Link Projection.</td>
<td>If a JourneyPatternTimingLink references a RouteLink, the start and end stops of both links should correspond. If the Direction of the JourneyPatternTimingLink is the same as that of the RouteLink, the respective start points should be the same and the respective ends point should be the same. If the Direction is opposite, the JourneyPatternTimingLink start point should match the RouteLink end point, and vice versa.</td>
<td>Int 1, p Reject</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jpt4</td>
<td>Start and end activity of journey pattern timing link.</td>
<td>Start activity of first stop of a JourneyPattern should be pickup only; activity of last stop should be set down. Unless route is circular, or stop connects at a JourneyPatternInterchange.</td>
<td>Int 6 Assume.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jpt5</td>
<td>Fare stages consistent with zone numbers.</td>
<td>The FareStage flag on stop usage of a From stop usage element should be set to reflect any change in FareStage zone numbers.</td>
<td>Int 6 Assume zone numbers are correct.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 14-3 – Intrinsic & Extrinsic Semantic Integrity Rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
<th>Integrity Level</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jptl6</td>
<td>Run Time should be greater than zero. Only in exceptional cases (e.g. physically adjacent stops) should a timing link run time be zero.</td>
<td>Int</td>
<td>6</td>
</tr>
<tr>
<td>Vj1</td>
<td>Cyclic vehicle journey references.</td>
<td>Int</td>
<td>3, p</td>
</tr>
<tr>
<td>Vj2</td>
<td>Vehicle journey link references. If a VehicleJourney references a VehicleJourney for its link usage, there should be no VehicleJourneyTimingLink instances present for the referencing journey.</td>
<td>Int</td>
<td>3, p</td>
</tr>
<tr>
<td>Vj3</td>
<td>Mixed Frequency Group</td>
<td>Int</td>
<td>3, p</td>
</tr>
<tr>
<td>Vj4</td>
<td>Vehicle journey direction. Vehicle journey Direction should be same as the journey pattern Direction.</td>
<td>Int</td>
<td>6</td>
</tr>
<tr>
<td>Vj5</td>
<td>Conflicting Frequency Group</td>
<td>Int</td>
<td>3, p</td>
</tr>
<tr>
<td>Vj1</td>
<td>Distinct interchange references. Inbound and outbound vehicle journeys of an interchange should be distinct.</td>
<td>Int</td>
<td>3, q</td>
</tr>
<tr>
<td>Vj2</td>
<td>Matching interchange journeys. The vehicle journeys referenced by a VehicleJourneyInterchange should be dependents of the corresponding inbound and outbound journey patterns referenced by the JourneyPatternInterchange that the VehicleJourneyInterchange references.</td>
<td>Int</td>
<td>3</td>
</tr>
<tr>
<td>Vj1</td>
<td>Vehicle journey timing link projection. For each VehicleJourneyTimingLink there should be a corresponding JourneyPatternTimingLink.</td>
<td>Int</td>
<td>1</td>
</tr>
<tr>
<td>Vj2</td>
<td>Start and end activity of vehicle journey timing link. Start activity of first stop of a VehicleJourney should be pickup only; activity of last stop should be set down. Unless route is circular, or stop connects at a VehicleJourneyInterchange.</td>
<td>Int</td>
<td>3, q</td>
</tr>
<tr>
<td>Vjp1</td>
<td>Positioning link distinct endpoints. From and to points of a positioning link should be distinct.</td>
<td>Int</td>
<td>3, q</td>
</tr>
<tr>
<td>Vjp2</td>
<td>Positioning link stop point. One end of a positioning link sequence should reference a stop in the journey pattern.</td>
<td>Int</td>
<td>3, p</td>
</tr>
<tr>
<td>Vjp3</td>
<td>Positioning link reference. Positioning link references should be valid. Any GarageRef instances referenced by a positioning link should belong to the Service Operator. Any GarageRef, LayoverRef instances referenced by a positioning link should belong to the JourneyPattern.</td>
<td>Int</td>
<td>3</td>
</tr>
</tbody>
</table>

### 14.3 Ordered Relationships

Table 14-4 shows the relationships in TransXChange whose order is semantically significant.
14.4 Precedence Rules for Combining General Date Elements

Table 14-5 shows the elements governing service dates, in order of precedence. Where elements cover the same day types or date ranges, higher precedence elements are used in preference to lower precedence elements. Data conflicts that are considered validation errors are indicated in a few cases.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Element</th>
<th>Description</th>
<th>Effect</th>
<th>Error</th>
<th>Sev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Service Period</td>
<td>Po1 Service / OperatingPeriod</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Vehicle Journey Interchange</td>
<td>Vi1 VehicleJourney / VehicleJourneyInterchange / ValidityPeriod, Exclude</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Vehicle Journey Special</td>
<td>Vx1 VehicleJourney / OperationProfile / SpecialDaysOperation / DaysOfNonOperation, Exclude</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Vx2 VehicleJourney / OperationProfile / SpecialDaysOperation / DaysOfOperation, Include</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Vx3 VehicleJourney / OperationProfile / BankHolidayOperation / DaysOfNonOperation</td>
<td>Exclude</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Vx4 VehicleJourney / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation</td>
<td>Include</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Vn2 VehicleJourney / OperationProfile / ServicedOrganisationDayType, DaysOfOperation</td>
<td>Include</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Vn3 VehicleJourney / OperationProfile / ServicedOrganisationDayType, of ServicedOrganisation / DaysOfNonOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>Exclude</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Vn4 VehicleJourney / OperationProfile / ServicedOrganisationDayType, of ServicedOrganisation / DaysOfOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>Include</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Vn5 VehicleJourney / OperationProfile / PeriodicDayType / WeekOfMonth.</td>
<td>Exclude</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Vn6 VehicleJourney / OperationProfile / RegularDayType / Days.</td>
<td>Include</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Journey Pattern Special</td>
<td>Jx1 JourneyPattern / OperationProfile / SpecialDaysOperation / DaysOfNonOperation, Exclude</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Jx2 JourneyPattern / OperationProfile / SpecialDaysOperation / DaysOfOperation, Include</td>
<td>T4 Outside of Service / OperatingPeriod</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Jx3</td>
<td>JourneyPattern / OperationProfile / BankHolidayOperation / DaysOfNonOperation.</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>Jx4</td>
<td>JourneyPattern / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation.</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>Journey Pattern Normal</td>
<td>Jn1</td>
<td>JourneyPattern / OperationProfile / ServicedOrganisationDayType / DaysOfNonOperation</td>
<td>exclude</td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td>Jn2</td>
<td>JourneyPattern / OperationProfile / ServicedOrganisationDayType, DaysOfOperation</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td>Jn3</td>
<td>JourneyPattern / OperationProfile / ServicedOrganisationDayType ServicedOrganisation / DaysOfNonOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td>Jn4</td>
<td>JourneyPattern / OperationProfile / ServicedOrganisationDayType, of ServicedOrganisation / DaysOfOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td>Jn5</td>
<td>JourneyPattern / OperationProfile / PeriodicDayType / WeekOfMonth.</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>Vn6</td>
<td>JourneyPattern / OperationProfile / RegularDayType / Days.</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td>Journey Pattern Interchange</td>
<td>Jt1</td>
<td>Service / JourneyPatternInterchange / ValidityPeriod, outside of range of Service / OperatingPeriod</td>
<td>exclude</td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td>Service Profile</td>
<td>Sx1</td>
<td>Service / SpecialOperationProfile / SpecialDaysOperation / DaysOfNonOperation</td>
<td>exclude</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>Sx2</td>
<td>Service / SpecialOperationProfile / SpecialDaysOperation / DaysOfOperation</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td>Sx3</td>
<td>Service / SpecialOperationProfile / BankHolidayOperation / DaysOfNonOperation</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>Sx4</td>
<td>Service / SpecialOperationProfile / BankHolidayOperation / DaysOfOperation</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>Service Normal Profile</td>
<td>Sn1</td>
<td>Service / OperationProfile / ServicedOrganisationDayType / DaysOfNonOperation</td>
<td>exclude</td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>Sn2</td>
<td>Service / OperationProfile / ServicedOrganisationDayType, / DaysOfOperation</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>Sn3</td>
<td>Service / OperationProfile / ServicedOrganisationDayType, of ServicedOrganisation / DaysOfNonOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td>Sn4</td>
<td>Service / OperationProfile / ServicedOrganisationDayType, of ServicedOrganisation / DaysOfOperation for the serviced organisations ancestors, as specified by ServicedOrganisation / ParentRef.</td>
<td>include</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td>Sn5</td>
<td>Service / OperationProfile / PeriodicDayType / WeekOfMonth.</td>
<td>exclude</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td>Sn6</td>
<td>Service / OperationProfile / RegularDayType / Days.</td>
<td>include</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 14-5 – Date Elements in Order of Precedence
APPENDIX A – REFERENCES TO OTHER STANDARDS

15.1 Transport Domain

15.1.1 NaPTAN & NPTG
National Public Transport Access Nodes (NaPTAN) database; NaPTAN seeks to assemble and maintain a single source of information on the location and naming of bus stops and other public transport access nodes in England, Wales and Scotland. [http://www.traveline.org.uk]

| UK Department for Transport NaPTAN & NPTG Schema User Guide 2.5 | 2013 March | Trapeze |
| http://www.dft.gov.uk/naptan/schema/2.5/NaPTAN.xsd | 2013 March | Trapeze |

15.1.2 JourneyWeb
JourneyWeb is a UK Department for Transport sponsored protocol which defines a national data standard for the dynamic interchange of transport information, including journey plans, and timetables. It is used by the Transport Direct Portal project.

| UK Department for Transport JourneyWeb Schema User Guide 2.5 | 2013 Jan | Trapeze |
| http://www.dft.gov.uk/journeyweb/schema/2.5/JourneyWeb2.5.xsd | 2013 Jan | Trapeze |

15.1.3 Transmodel CEN TC 278
Transmodel is a CEN sponsored abstract standard for describing Public Transport Information Systems.


15.1.4 Netex CEN TC 278
Transmodel is a CEN sponsored conceptual model and XML schema based on Transmodel for describing Public Transport Information Systems

| CEN | 2013 Jan | CEN TC* &* CEN TC* &* |
| http://www.netex.org.uk | 2013 Jan | CEN TC278 SG9 |

15.2 Software & General

15.2.1 XML Schema


15.2.2 ISO Time Formats

| ISO 8601 Date and Time Formats. http://www.w3.org/TR/iso8601 | 2001 May 2 | W3C Various |
## ISO8601:2000(E) - Data elements and interchange formats – Information interchange – Representation of dates and times

Second edition 2000-12-15


2000 Dec 15

Louis Visser

### 15.2.3 WGS 1984 Location Referencing

World Geodetic Standard 1984


W3C Various

### 15.2.4 ISO 639-1 Names of Languages


http://www.oasis-open.org/cover/iso639a.html

Infoterm

### 15.2.5 Rfc 1766 Tags for the Identification of Languages

rfc1766 – Tags for the Identification of Languages

http://www.ietf.org/rfc/rfc1766.txt

Infoterm

### 15.2.6 GovTalk XML Coding Standards

GovTalk sets out standards for exchange of data in XML

Office of the e-Envoy

Schema Guidelines

Best Practice Advice Version 2

Schema Guidelines 2.doc

2002 Oct 12

Paul Spencer

e-Government Metadata Standard
e-GMS1.0

-Government_Metadata_Standard_v1.pdf

2002 Apr

Office of e-Envoy

### 15.2.7 UML Unified Modelling Language

Unified Modelling Language is a notation for describing software models managed by the Object Management Group.

Unified Modelling Language (UML), version 1.5


formal/2003-03-01

OMG