

UK Public Transport Information Profile

Versioning in TxC-PTI


Version 1.0

24 March 2022

This application note on how to use versioning in the TxC-PTI should be read in conjunction with the 'TransXChange UK PTI Profile v1.1.A' document for the wider context and use of TxC.

This advice supersedes the contents of Section 2.3 of the 'TransXChange UK PTI Profile v1.1.A' document.

Updates to the text in section 2.3 are identified using **blue** text. These changes are to provide additional clarity on how data is expected to be provided using the profile.

Only one of the updates, that in the section identified by the  mandatory symbol changes the requirements of the profile.

Section 2.3.1 is substantially updated and should be re-read in its entirety to ensure proper understanding.

Section '2.3.2 Multiple TransXChange Files' is a new section.

2.3 Versions

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

TXC-PTI has adopted some general rules about versioning which shall be followed in those elements which support it. The general principle shall be that, wherever possible:

- **ids and codes used within a series of documents to describe a service shall be consistent from document to document, for example ServiceCode (see 5.3.2), and**
- **that sequences of documents shall be identified and tracked through the documents' version control attributes.**

The image shows a screenshot of a document titled 'attributes' with a list of element attributes. Each attribute is enclosed in a dashed box and followed by its description. The attributes listed are: id, CreationDateTime, ModificationDateTime, Modification, RevisionNumber, Status, BaselineVersion, DataRightRef, and layer.

- id**: Identifier of element
- CreationDateTime**: Timestamp at creation of entity. Should be set when the entity is first created, and not subsequently be changed.
- ModificationDateTime**: Date of most recent update. Should be changed every time an entity is changed, or when any of its child entities that are not themselves versioned are changed. May be omitted if Modification is new, i.e. if same as CreationDateTime, otherwise must be specified. Will be equal or later than the CreationDateTime
- Modification**: Nature of data change of exchanged entity:
 - New: This is the first version of the element instance, as created for the first time. An entity continues to have a status of new until it is revised. The creation date can be used to detect a recent addition.
 - Revise: This is an update to an existing element instance, or any of its child elements are being updated, added, or deleted. Once an element is marked as revise it will continue to be so unless it is marked as deleted, i.e. should not ever revert to new. If no value is specified, revise will be assumed.
 - Delete: The element is being rendered inactive. Records marked as deleted should continue to be exported in subsequent data exchanges but is deprecated 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.
- RevisionNumber**: The RevisionNumber 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 Issuer should increment this numberThe RevisionNumber of an instance should not be changed if there is no change to the data values or children of an element.
- Status**: Status of entity at time of export. Indicates whether after the modification the element will be considered to be active, inactive, or pending, (i.e. inactive subject to verification)
 - 'Active': Entity is either in use or available to be used.
 - 'Inactive': Entity is in database but is marked as ' .
 - 'Pending': Entity is missing, or flagged as deleted from the most recent data upload, and may be in process of being made inactive.
- BaselineVersion**: Baseline version with which this version is compatible. (+NaPTAN v2.4)
- DataRightRef**: Reference to a Data right that governs the use of this element and its children. (+NaPTAN v2.4)
- layer**: Data layer to which this element is assigned. Metadata (+NaPTAN v2.4).

Figure 1 - An indicative set of element attributes

- **CreationDateTime** shall be populated with the date and time at which the **data element** was first created and shall not be changed in any future version of the same element.
- **ModificationDateTime** shall be populated with the date and time **that the revision was created**. It shall be omitted for the original (version 0) issue of the data element but shall be present in all subsequent versions of the same **data element**.

- *Modification* is a status flag and shall be set to either “New” for the first issue of the data element, or to “Revise” for subsequent versions of the same **data element**. No other *Modification* enumerators shall be used.
- *RevisionNumber* is a sequence number that shall be 0 for the first issue of the data element and that shall increase for each subsequent version of the **data element**. It is important to note that this revision number is used to track TXC-PTI data from version to version and is independent of any other versioning system. **It should be noted that the Service RevisionNumber in TXC-PTI may change for reasons that do not need to be registered with the Traffic Commissioner, and therefore the Service RevisionNumber will rarely if ever match the variation number of the registration..**
- *Status* is of limited value since data elements (e.g. **Service/OperatingProfile**) will describe the extent to which data is active. There is therefore no requirement to provide this attribute, and for any conflicts between *Status* and data, the *Status* flag shall be ignored.

These are summarised in the table below:

Table 1 - Attributes used for versioning

Attribute Name	Data Type	Used in TXC-PTI
CreationDateTime	<i>dateTime</i>	Yes. Always present and does not change. The date and time of revision number 0.
ModificationDateTime	<i>dateTime</i>	Yes, for all revision numbers other than 0. Omitted otherwise. The date and time this revision was created.
Modification	<i>ModificationEnumeration</i>	Yes. Permitted values are: <ul style="list-style-type: none"> • New • Revise
RevisionNumber	<i>RevisionNumberType</i>	Yes. First version is revision number 0. Increment by 1 for each revision thereafter.
Status	<i>StatusEnumeration</i>	Optional



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

Where journeys of the same Service are presented in different files each file must have the same ServiceCode and RevisionNumber.



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



The validator tests, at a file level for:

- CreationDateTime being always present and that it remains the same between files submitted with different RevisionNumber.

- ModificationDateTime being newer than CreationDateTime where RevisionNumber is greater than 0.
- Modification is a valid value; either New or Revise.
- RevisionNumber for new revisions is larger than the existing published revision.

Using RevisionNumber in BODS as a data consumer:

- When a new file is supplied to BODS with a new RevisionNumber the previous RevisionNumber file shall be superseded from the start of the OperatingPeriod of the new file or Service ModificationDateTime (which for these purposes will always be in the past) if OperatingPeriod is unchanged from the previous RevisionNumber.
- New higher RevisionNumber files supersede lower RevisionNumber files in their entirety.

2.3.1 Versioning in BODS



Examples of the use of RevisionNumber for versioning in PTI-TxC are provided below.

Fundamentally the approach to revisioning and superseding files and data elements within BODS reflects the approach taken by the local bus service registration process. Start and end dates are configured using Service/OperatingPeriod.

The initial file provided to BODS for a service would have a RevisionNumber = 0, in this example the file is published on 1st December 2021 with the service starting on 1 January 2022 operating until further notice (no end date) . As a result of only a single Service being allowed in a PTI-TXC the TxC document RevisionNumber and Service RevisionNumbers will always match.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	...
0	01/12/2021	01/01/2022							

When there is a change to the service a new file is provided to BODS on 1st January 2022 with a RevisionNumber = 1. This change starts on 1 February 2022.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	...
1	01/01/2022	01/02/2022							

When consuming the file on 5 January 2022 the file will be read and the OperatingPeriod StartDate is 1st February 2022. There is no valid data until 1st February 2022.

To update data for a Service in advance of a change and to retain the previous operating information two files need to be published to BODS, provided in a ZIP file. This ZIP file is published to BODS on 1st January 2022.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	...
0	01/12/2021	01-Jan-22							
1	01/01/2022	01-Feb-22							

Both files RevisionNumber = 0 and RevisionNumber = 1 will be available to data consumers.

Data consumers will download and import the contents of the ZIP file from BODS. Upon importing it is identified that whilst the file with RevisionNumber = 1 has the greatest RevisionNumber the OperatingPeriod StartDate in the file is not valid until 1st February 2022. The data consumer then checks the file with RevisionNumber = 0 and identifies that this data is valid on 5th January. On the 1st February 2022 the RevisionNumber = 1 file becomes valid and supersedes the file RevisionNumber = 0.

The RevisionNumber = 0 file will continue to be available along with RevisionNumber = 1 until a new ZIP file that does not include the RevisionNumber = 0 file is published.

In the situation where there is a short term change to a Service a file with a new RevisionNumber needs to be published.

In this example RevisionNumber = 1 is a service running from 1st February 2022 with no end date. A short duration timetable operating from 1st March 2022 to 31st March 2022 is provided as RevisionNumber =2.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	...
1	01/01/2022	01/02/2022	31/12/2022						...
2	01/02/2022	01/03/2022	31/03/2022						...

On 1st April 2022 the end date of RevisionNumber = 2 has passed, the file with RevisionNumber = 1 is still within the ZIP file, but because the RevisionNumber has been superseded by RevisionNumber = 2 and the End Date has passed there is no valid data.

If the short duration timetable reverts to the previous timetable after the end date, then the previous timetable data needs to be re-supplied with a new OperatingPeriod start date, and an incremented Service RevisionNumber.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	...
2	01/02/2022	01/03/2022	31/03/2022						...
3	01/02/2022	01/04/2022							...

If the Service is ending and will not operate in future, then a new file with an end date needs to be provided.

Service RevisionNumber	Published Date	Start Date	End Date	Dec 2021	Jan 2022	Feb 2022	March 2022	April 2022	May 2022
4	01/04/2022	01/04/2022	30/04/2022						

In this example after 30th April 2022 there is no valid data and the service can be interpreted as ended.

In addition to single Service and Line data sets the PTI-TxC profile supports multiple Lines within a single Service.

In this example a new Service is registered for the first time published on 1st February 2022 with a start date of 1st February 2022. This is a new service so ModificationDateTime = CreationDateTime and Modification = New.

Service	Service RevisionNumber	Line	Line RevisionNumber	CreationDateTime	ModificationDateTime	Modification	Comment
4	0			2022-02-01T18:53:49	2022-02-01T18:53:49	New	New registration
		4a	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	Initial timetable
		4b	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	Initial timetable

Service	Service RevisionNumber	Line	Line RevisionNumber	Modification Date	Start Date	EndDate	Feb 2022	March 2022	April 2022	...
4	0	4a	0	01/02/2022	01/02/2022				...	
4	0	4b	0	01/02/2022	01/02/2022				...	

If the Service is being amended to add an additional line, Line 4c, a new file is provided including the additional Line. No matter if different Lines appear in the same file or in separate files, the Service has changed and therefore the Service RevisionNumber and also the file/document RevisionNumber need to be incremented for all files relating to that Service. For Lines that were in the Service previously no change is required.

Service	Service RevisionNumber	Line	Line RevisionNumber	CreationDateTime	ModificationDateTime	Modification	Comment
4	1			2022-02-01T18:53:49	2022-02-14T15:29:25	Revise	Revision Number increased because new file created to supply new line 4c
		4a	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	No change
		4b	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	No change
		4c	0	2022-02-14T15:29:25	2022-02-14T15:29:25	New	New Line 4c added

Service	Service RevisionNumber	Line	Published Date	Start Date	End Date	Feb 2022	March 2022	April 2022	...
4	1	4a	14/02/2022	01/02/2022					...
4	1	4b	14/02/2022	01/02/2022					...
4	1	4c	14/02/2022	01/03/2022					...

To remove a Line from a service, in this case Line 4b is removed from the service, a new file is provided removing the Line. The Service RevisionNumber is increased but because there is no change to the remaining Lines these are not updated.

Service	Service RevisionNumber	Line	Line RevisionNumber	CreationDateTime	ModificationDateTime	Modification	Comment
4	2			2022-02-01T18:53:49	2022-02-15T15:29:25	Revise	Revision Number increased because new file created to remove line 4b
		4a	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	No change
		4c	0	2022-02-14T15:29:25	2022-02-14T15:29:25	New	No change

This will remove the Line immediately as RevisionNumber has been increased and superseded all previous data.

Service	Service RevisionNumber	Line	Line RevisionNumber	Line Modification Date	Start Date	EndDate	Feb 2022	March 2022	April 2022	...
4	2	4a	0	01/02/2022	01/02/2022					...
4	2	4c	0	14/02/2022	01/03/2022					...

To remove line 4b at a future date then the Service needs to have its OperatingPeriod EndDate set this would result in the Line RevisionNumber increased along with the Service RevisionNumber. The unaffected Lines RevisionNumbers remain unchanged.

Service	Service RevisionNumber	Line	Line RevisionNumber	CreationDateTime	ModificationDateTime	Modification	Comment
4	3			2022-02-01T18:53:49	2022-02-16T10:16:27	Revise	Revision Number increased because updated line 4b
		4a	0	2022-02-01T18:53:49	2022-02-01T18:53:49	New	No change
		4b	1	2022-02-01T18:53:49	2022-02-16T10:16:27	Revise	OperatingPeriod end date added
		4c	0	2022-02-14T15:29:25	2022-02-14T15:29:25	New	No change

Service	Service RevisionNumber	Line	Line RevisionNumber	Line Modification Date	Start Date	EndDate	Feb 2022	March 2022	April 2022	...
4	2	4a	0	01/02/2022	01/02/2022					...
4	2	4b	1	16/02/2022	01/02/2022	31/03/2022				...
4	2	4c	0	14/02/2022	01/03/2022					...

RevisionNumbers within a TxC file are useful for tracking changes to data elements that comprises a Service. For example in the Service 4b example above allows a data consumer to identify that two of the three Lines are unaffected. This knowledge is important for customer information about changes, understanding what bus stop information may need to be updated etc.

2.3.2 Multiple TransXChange Files

Within a Service where multiple TransXChange files are supplied for a Service the RevisionNumbers within each file must match for valid data. For example where separate files are provided for Monday to Friday and Saturday journeys RevisionNumber must match.

Where data for a Service is supplied in multiple files these must be in a ZIP file and data consumers need to import all files to ensure completeness.

In this example the data for the Service is supplied in three files A, B and C. The ZIP available from BODS on 14th February 2022 has all three files with RevisionNumber = 1.

File	LineId	Contents	Service RevisionNumber at 14/02/2022
A	CSLB:PF1000671:63:36	M – F timetable WorkingDays	1
B	CSLB:PF1000671:63:36	M – F timetable NonWorkingDays	1
C	CSLB:PF1000671:63:36	Saturday	1

A new ZIP file is then supplied with files A and B having RevisionNumber = 2 and file C RevisionNumber =1.

File	LineId	Contents	Service RevisionNumber at 14/02/2022	Service RevisionNumber at 28/02/2022
A	CSLB:PF1000671:63:36	M – F timetable WorkingDays	1	2
B	CSLB:PF1000671:63:36	M – F timetable NonWorkingDays	1	2
C	CSLB:PF1000671:63:36	Saturday	1	1

These files are for the same Service and therefore file C having a RevisionNumber = 1, which is less than the RevisionNumber used for files A and B, should be regarded as superseded and no longer valid. In this example this would remove the Saturday journeys.