



## **THE INSTITUTE OF NAVIGATION**

8551 Rixlew Lane, Ste. 360

Manassas, VA 20109-3701

PHN: (703) 366-2723/FAX: (703) 366-2724

[WWW.ION.ORG](http://WWW.ION.ORG)

# **Global Navigation Satellite Systems**

## **Software Defined Radio**

### **Sampled Data**

### **Metadata Standard**

Revision ~~1~~2.0

ION GNSS SDR Standard Working Group

#### **Abstract**

The Global Navigation Satellite Systems (GNSS) Software Defined Radio (SDR) Metadata Standard defines parameters and schema to express the contents of SDR sampled data files. The standard is designed to promote the interoperability of GNSS SDR data collection systems and processors. The standard includes a formal XML schema definition (XSD). A compliant open source C++ applications programming interface (API) is also officially supported to promote ease of integration into existing SDR systems.

## Table of Contents

1	Introduction.....	8
2	Scope.....	8
3	Metadata Format .....	8
4	SDR Data Collection Topologies.....	8
4.1	Single Band, Single Stream, Single File / Multiple Files .....	10
4.2	Multi-Band, Single Stream, Single File .....	10
4.3	Multi Stream, Single File .....	11
4.4	Multi Stream, Single File (with Additional Data).....	11
4.5	Temporal Splitting of Files .....	11
4.6	Spatial Splitting of Files.....	12
4.7	Spatial-Temporal Splitting of Files.....	13
5	Metadata File Naming and Association Mechanisms .....	13
6	Domain Model .....	13
6.1	Architecture.....	15
6.2	Core Classes.....	18
6.2.1	Session object.....	18
6.2.2	System object .....	19
6.2.3	Cluster object .....	19
6.2.4	Source object.....	21
6.2.5	Band object .....	22
6.2.6	Stream object .....	23
6.2.7	Lump object .....	27
6.2.8	Chunk object .....	29
6.2.9	Block object .....	31
6.2.10	Lane object.....	33
6.2.11	File object.....	34
6.2.12	FileSet object .....	36
6.3	Foundation Classes .....	37
6.3.1	URI.....	37
6.3.2	DateTime.....	37
6.3.3	Frequency .....	37
6.3.4	Duration .....	37

6.3.5	Position .....	38
6.3.6	Origin .....	38
6.3.7	Orientation .....	39
6.3.8	String .....	39
Appendix I: Encoding Function .....		40
Appendix II: Future Extensions .....		42
1	Introduction .....	6
2	Scope .....	6
3	Metadata Format .....	6
4	SDR Data Collection Topologies .....	6
4.1	Single Band, Single Stream, Single File / Multiple Files .....	7
4.2	Multi-Band, Single Stream, Single File .....	7
4.3	Multi-Stream, Single File .....	8
4.4	Multi-Stream, Single File (with Additional Data) .....	8
4.5	Temporal Splitting of Files .....	8
4.6	Spatial Splitting of Files .....	9
4.7	Spatial-Temporal Splitting of Files .....	9
5	Metadata File Naming and Association Mechanisms .....	9
6	Domain Model .....	9
6.1	Architecture .....	10
6.2	Core classes .....	11
6.2.1	Session object .....	11
6.2.2	System object .....	12
6.2.3	Cluster object .....	12
6.2.4	Source Object .....	13
6.2.5	Band object .....	14
6.2.6	Stream object .....	14
6.2.7	Lump object .....	17
6.2.8	Chunk object .....	18
6.2.9	Block object .....	19
6.2.10	Lane object .....	20
6.2.11	File object .....	21
6.2.12	FileSet Object .....	22
6.3	Foundation Classes .....	22
GNSS SDR Metadata Standard. Rev 12.0 Aug 14 2017 Jul 23 2018		

6.3.1	URI .....	22
6.3.2	DateTime .....	22
6.3.3	Frequency .....	22
6.3.4	Duration .....	22
6.3.5	Position .....	23
6.3.6	Origin .....	23
6.3.7	Orientation .....	24
7	Appendix: .....	25
	Encoding Functions .....	25

Formatted  
(United S

Formatted  
Formatted  
Formatted  
Formatted  
Formatted

## List of Figures

Figure 1 — Fundamental GNSS SDR Data Collection Topologies .....	10
Figure 2 — Overview of Core Metadata Classes and Generation .....	14
Figure 3 — GNSS metadata class model (UML 2.0).....	16
Figure 4 — Core metadata classes specialize the base metadata element, which has a unique identifier (id), links to related artifacts (URI) and comments .....	17
Figure 5 — Intentional Aliasing of a Multiband signal to Baseband .....	23
Figure 6 — Illustration of Multiple Bands Present in a Stream .....	24
Figure 7 — Illustration of a lump containing samples from N streams.....	27
Figure 8 — Encoding schemes for a single lump within a single chunk .....	30
Figure 1 - Fundamental GNSS SDR data collection topologies .....	10
Figure 2 - Overview of core metadata classes and generation .....	14
Figure 3 - GNSS metadata class model (UML 2.0).....	16
Figure 4 - Core metadata classes specialize the base metadata element, which has a unique identifier (id), links to related artifacts (URI) and comments.....	17
Figure 5 - Intentional aliasing of a multiband signal to baseband .....	23
Figure 6 - Illustration of multiple bands present in a stream .....	24
Figure 7 - Illustration of a lump containing samples from N streams.....	27
Figure 8 - Encoding schemes for a single lump within a single chunk.....	30
Figure 9 - Encoding schemes for N chunks within a block .....	32
Figure 10 - Encoding scheme for N blocks within a lane .....	34
Figure 11- Encoding of the lanes within a file .....	36

## List of Tables

Table 1 — Metadata element class attributes .....	18
Table 2 — Definition of session attributes .....	18
Table 3 — Definition of system attributes .....	19
Table 4 — Definition of cluster attributes .....	20
Table 5 — Definition of source attributes .....	21
Table 6 — Definition of band attributes .....	23
Table 7 — Definition of stream attributes .....	25
Table 8 — Enumeration of stream encoding attribute .....	26
Table 9 — Definition of lump attributes .....	28
Table 10 — Definition of chunk attributes .....	29
Table 11 — Definition of block attributes .....	32
Table 12 — Definition of lane attributes .....	33
Table 13 — Definition of file attributes .....	35
Table 14 — Definition of fileSet attributes .....	36
Table 15 — Definition of position attributes .....	38
Table 16 — Definition of origin attributes .....	38
Table 17 — Definition of orientation attributes .....	39
Table 18 — Encoding of 2-bit samples .....	40
Table 19 — Encoding of 3-bit samples .....	40
Table 20 — Encoding of 4-bit samples .....	40
Table 21 — Encoding of 5-bit samples .....	41
Table 1 - Metadata element class attributes .....	18
Table 2 - Definition of session attributes .....	18
Table 3 - Definition of system attributes .....	19
Table 4 - Definition of cluster attributes .....	20
Table 5 - Definition of source attributes .....	21
Table 6 - Definition of band attributes .....	23
Table 7 - Definition of stream attributes .....	25
Table 8 - Enumeration of stream encoding attribute .....	26
Table 9 - Definition of lump attributes .....	28
Table 10 - Definition of chunk attributes .....	29
Table 11 - Definition of block attributes .....	32
Table 12 - Definition of lane attributes .....	33
Table 13 - Definition of file attributes .....	35
Table 14 - Definition of fileSet attributes .....	36
Table 15 - Definition of position attributes .....	38
Table 16 - Definition of origin attributes .....	38
Table 17 - Definition of orientation attributes .....	39
Table 18 - Encoding of 2-bit samples .....	40
Table 19 - Encoding of 3-bit samples .....	40
Table 20 - Encoding of 4-bit samples .....	40
Table 21 - Encoding of 5-bit samples .....	41

## List of acronyms

<u>ADC</u>	<u>Analog to digital converter</u>
<u>API</u>	<u>Applications programming interface</u>
<u>BPF</u>	<u>Band pass filter</u>
<u>DCS</u>	<u>Data collection system</u>
<u>ECEF</u>	<u>Earth-centered, earth-fixed</u>
<u>GNSS</u>	<u>Global navigation satellite system</u>
<u>GPS</u>	<u>Global positioning system</u>
<u>GTRF</u>	<u>Galileo terrestrial reference frame</u>
<u>id</u>	<u>Identifier</u>
<u>IF</u>	<u>Intermediate frequency</u>
<u>LHCP</u>	<u>Left handed circular polarization</u>
<u>LLH</u>	<u>Latitude, longitude, height</u>
<u>MSB</u>	<u>Most significant bits</u>
<u>PC</u>	<u>Personal computer</u>
<u>UML<sub>poc</sub></u>	<u>Unified model language</u> <u>Person of contact</u>
<u>URI<sub>ppm</sub></u>	<u>Universal resource identifier</u> <u>Parts per million</u>
<u>PZ-90</u>	<u>Parametry zemli 1990</u>
<u>RF</u>	<u>Radio frequency</u>
<u>RHCP</u>	<u>Right handed circular polarization</u>
<u>RTC</u>	<u>Real time clock</u>
<u>SDR</u>	<u>Software defined radio</u>
<u>SF</u>	<u>Sample file</u>
<u>toa</u>	<u>Time of applicability</u>
<u>UML</u>	<u>Unified model language</u>
<u>URI</u>	<u>Universal resource identifier</u>
<u>URL</u>	<u>Universal resource locator</u>
<u>UTF</u>	<u>Unicode transformation format</u>
<u>WGS</u>	<u>Word geodetic system</u>
<u>XML</u>	<u>Extensible mark-up language</u>
<u>XSD</u>	<u>XML schema definition</u>
<u>XML</u>	<u>Extensible markup language</u>





# 1 Introduction

The past several years has seen a proliferation of software defined radio (SDR) data collection systems and processing platforms that are particularly designed for Global Navigation Satellite System (GNSS) receiver applications or those that support GNSS bands. For post-processing, correctly interpreting the GNSS SDR sampled datasets produced or consumed by these systems has historically been a cumbersome and error-prone process. This is because these systems necessarily produce datasets of various formats, the subtleties of which are often lost in translation when communicating between the producer and consumer of these datasets. This specification standardizes the metadata description associated with GNSS SDR sampled data files.

## 2 Scope

Datasets containing GNSS SDR samples may also contain other information such as sensor data and data from radio frequency (RF) bands other than GNSS. For non-RF data, this specification supports bypassing this data during reading. For non-GNSS RF bands, only parameters common to GNSS bands are supported.

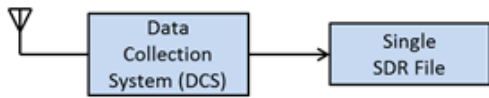
## 3 Metadata Format

Extensible ~~Markup~~Mark-up Language (XML) is used in this standard. The XML schema ~~is~~are specified according to the XML Schema Definition (XSD) standard.

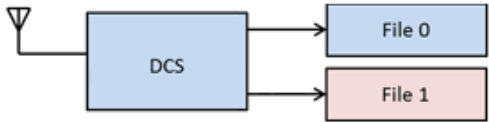
## 4 SDR Data Collection Topologies

This standard is designed to support most current and future GNSS SDR sampled data file formats. These formats stem from the fundamental data collection topologies illustrated in ~~Figure 1~~Figure 1. This section describes these topologies.

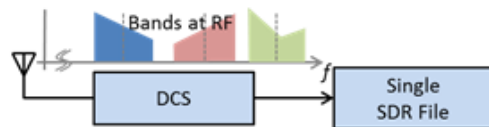
(a) single band, single-stream, single file



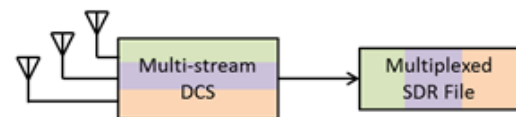
(b) single band, single-stream, multiple files



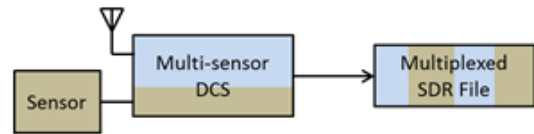
(c) multi-band, single stream, single file



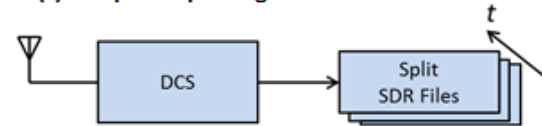
(d) multi-stream, single file



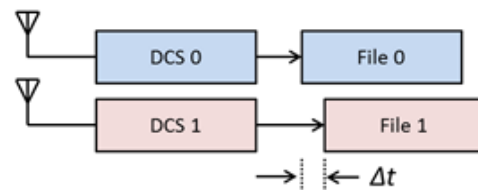
(e) multi-stream, single file



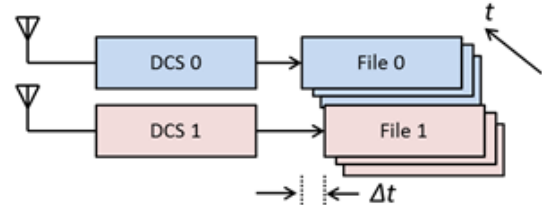
(f) temporal splitting of files



(g) spatial splitting of files



(h) spatial-temporal splitting



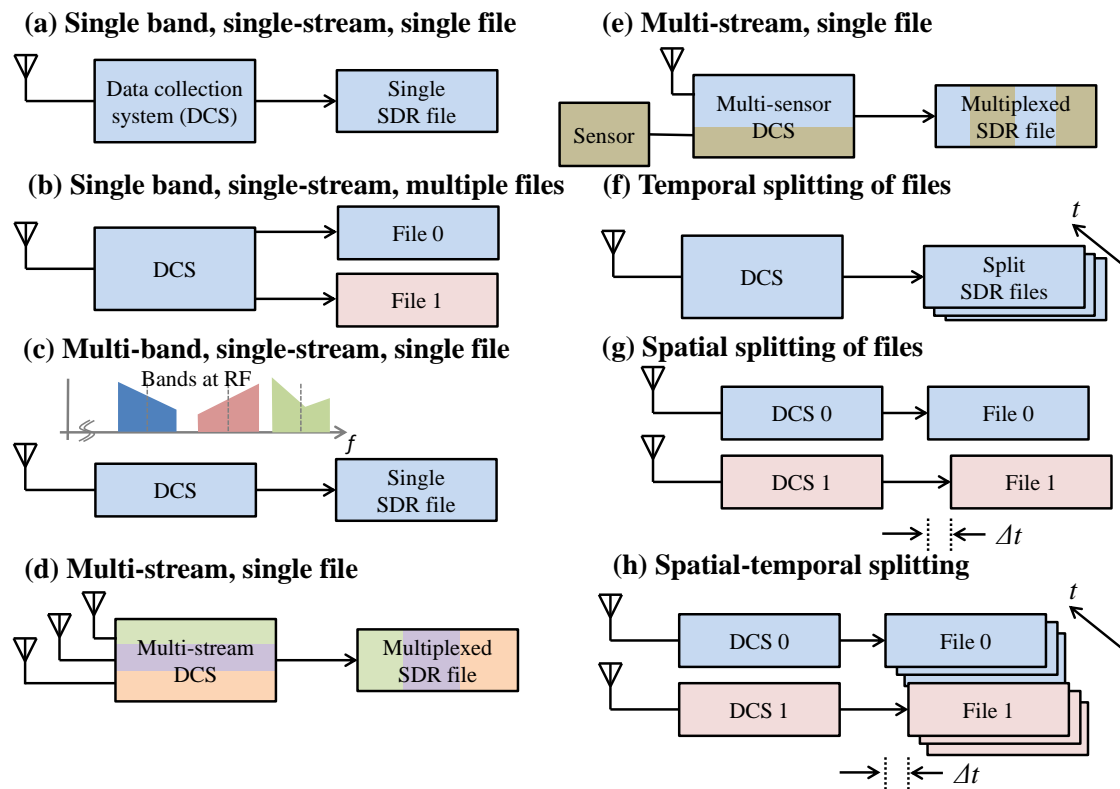


Figure 1 – Fundamental GNSS SDR Data Collection Topologies

### 3.14.1 Single Band, Single Stream, Single File / Multiple Files

Figure 1.a illustrates the simplest data collection topology that can exist. This is when a single contiguous region of RF spectrum (referenced henceforth as a ‘band’) is down-converted and sampled to produce a single data stream that is then written to a single data file.

For this and all subsequent topologies, the data stream may contain samples that are either real or complex valued depending on whether intermediate frequency (IF) or baseband sampling is used, respectively. These samples are packed according to a repetitive pattern. The repetitive pattern may also comprise of other information at the beginning and/or end of a fixed number of samples. This may include non-sample data such as headers and footers which, for example, may be used for data integrity check purposes. In this topology, this formatted data stream is written to one and only one file. However, some systems prefer to write the formatted data stream into several data files as shown in Figure 1.b.

### 3.24.2 Multi-Band, Single Stream, Single File

Figure 1.c is identical to Figure 1.a in terms of how the data stream may be formed and written to disk, except the data stream contains information from more than one RF band. An example of this topology is a direct RF sampling front-end architecture that intentionally aliases multiple bands such that they appear next to each other at baseband. In this case, some bands may be spectrally inverted as a result of the digital down-conversion process.

A similar topology can be achieved as in Figure 1.b, where the data stream containing various bands is saved in multiple files.

### 3.4.3 Multi Stream, Single File

**Figure 1.d** illustrates a topology where multiple sample streams are combined into a single formatted data stream and written to a single file. The formatted data stream may contain additional information as described in 4.1.4.1. Each sample stream represents a distinct time series that is independent from any and all others (i.e. independent in a mathematical time series sense, not in a statistical sense).

#### NOTE:

The distinction of sample stream (i.e. mathematical time series) versus data stream (i.e. formatted data bytes that are ultimately written to disk) is made above. In this standard, the term *stream* shall always imply the former. The term *data stream* shall be used specifically to refer to the latter.

In the example shown, each sample stream represents the data collected from a different antenna whose signal passes through a different RF front-end channel. This is for illustration purposes only. The standard does not assume any dependence between streams (including common sample rates or quantization).

### 3.4.4 Multi Stream, Single File (with Additional Data)

**Figure 1.e** illustrates a data stream containing GNSS samples as well as data from an additional sensor. For the purpose of this standard, any data that cannot be represented as GNSS sample streams are considered unknown data. The standard defines parameters necessary to skip over unknown data bytes when decoding the data stream.

The remaining topologies (**Figure 1.f** - **Figure 1.h**) address how a data stream may be written to disk.

### 3.5.5 Temporal Splitting of Files

The data rates of GNSS SDR streams are typically high (on the order of one to several hundred MB/secs). Hence, long-duration data collections can generate very large files that become cumbersome to manage. For this reason, the data may be written to smaller sets of files (illustrated in **Figure 1.f**) where the data stream continues from the end of one file to the beginning of another (possibly with some overlap to ensure data integrity). This is defined as *temporal file splitting* in this standard. The standard includes parameters that specify the chronological order of temporally split files.

#### NOTE:

A metadata file typically exists for each data file. Optionally, all information for a multi-file set may be contained within one metadata file. For the former case, the first metadata file of a set must contain or make reference to the complete set of metadata parameters and subsequent files may contain only those that change from file to file.

### 3.64.6 Spatial Splitting of Files

A collection system or setup may write individual data streams or the frequency bands to multiple files (illustrated in [Figure 1.g](#)). These files may be written within the same host system (such as a personal computer (PC)) or multiple systems. This is defined as *spatial file splitting* in this standard.

**NOTE:**

This standard associates two or more spatially split files in a specification defined as *fileSet*.

### 3.7.4.7 Spatial-Temporal Splitting of Files

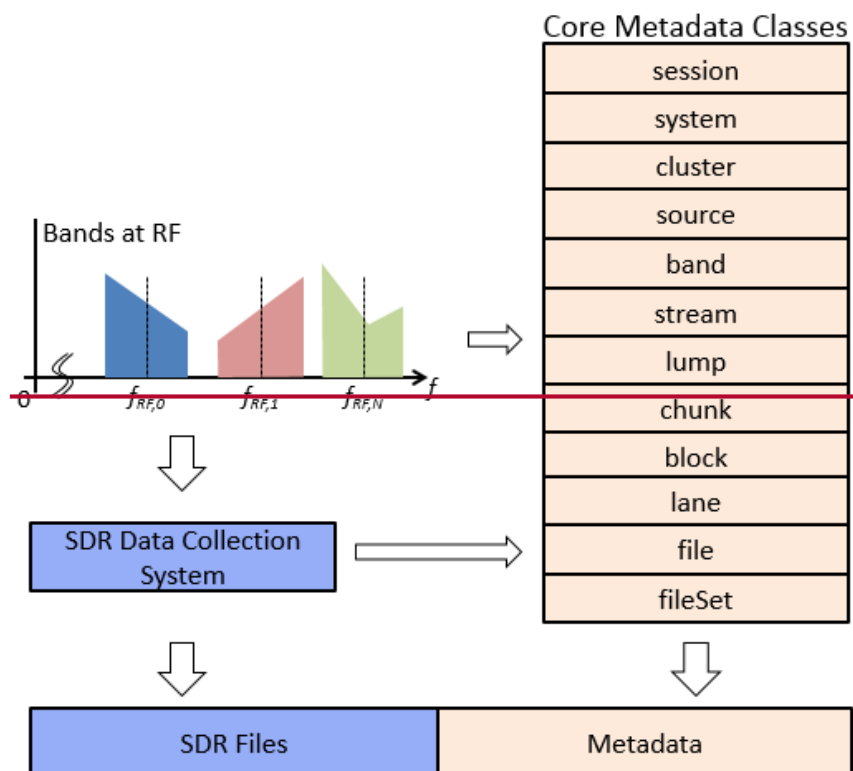
**Figure 1** Figure 1.h illustrates the combination of spatial and temporal splitting. In this case, the fileSet parameter refers to the first of each temporally split file.

## 4.5 Metadata File Naming and Association Mechanisms

The official filename extension for a metadata file is '.sdrx'. Use of this extension is recommended.

## 5.6 Domain Model

As illustrated in **Figure 2** Figure 2, metadata are defined in terms of 12 core classes, represented in the orange boxes. These core classes are explained in the different subsections within this chapter.



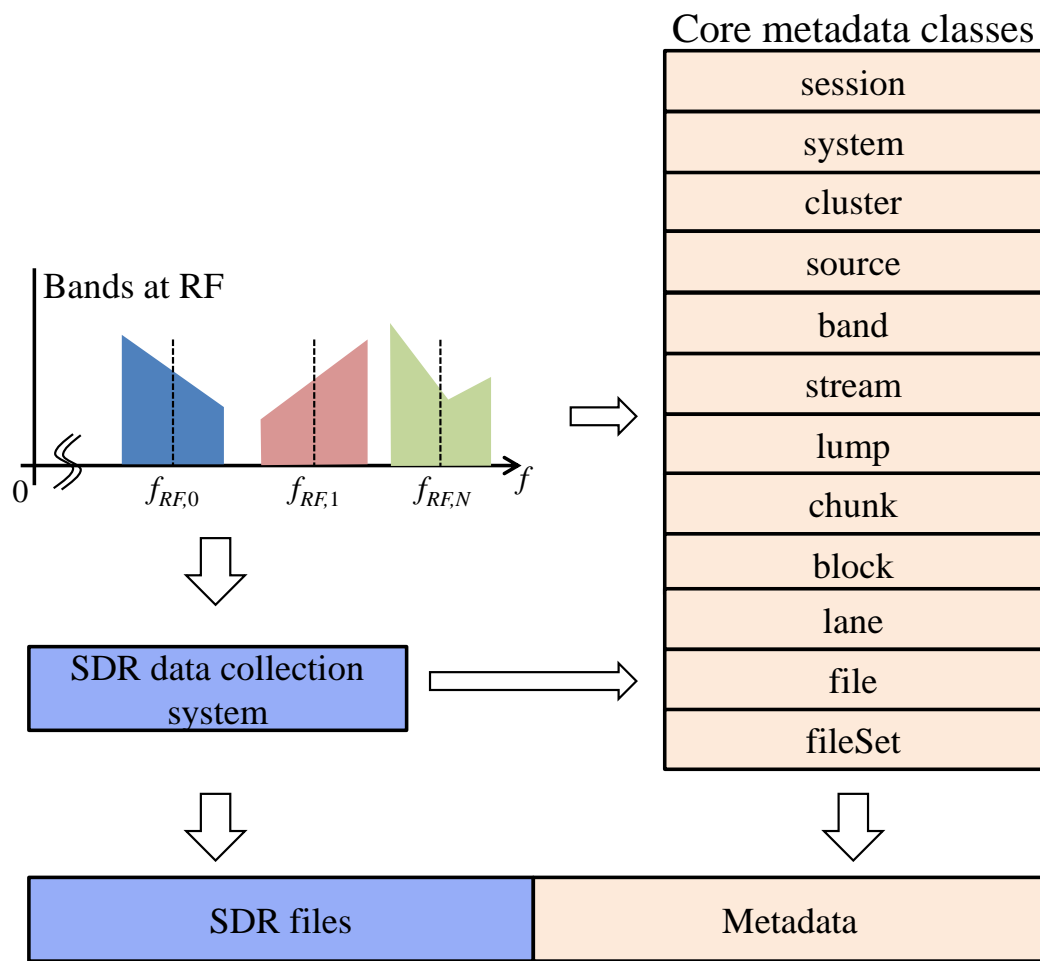
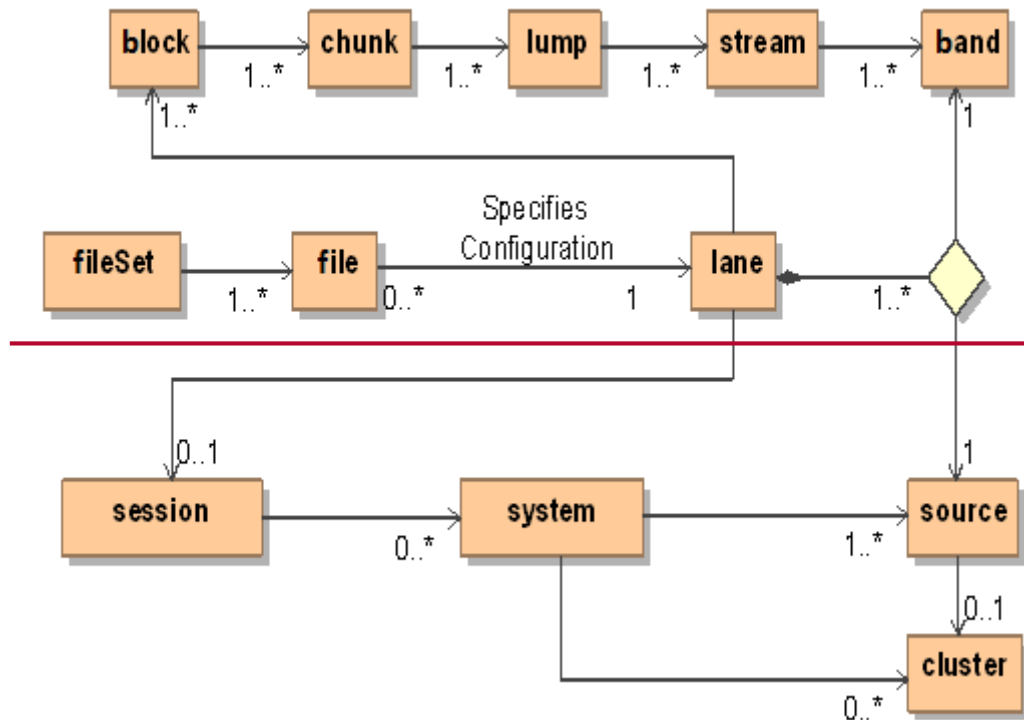


Figure 2 – Overview of ~~Core Metadata Classes~~core metadata classes and ~~Generation~~generation

### 5.16.1 Architecture

Figure 3 and Figure 4 show the relation of the different core classes. This relation is shown between the different core classes (Figure 3), and between the core classes and the main class (Figure 4), namely *metadata*.



The numeric values in Figure 3 (“1..\*”, “0..1” and “0..\*”) represent the possible number of core classes contained inside the previous core class, e.g. if a fileSet core class is defined, this will contain at least one (1..\*) file core class inside. The notation “0..\*” or “0..1” implies that there is no need to have the corresponding core class contained inside the previous core class.



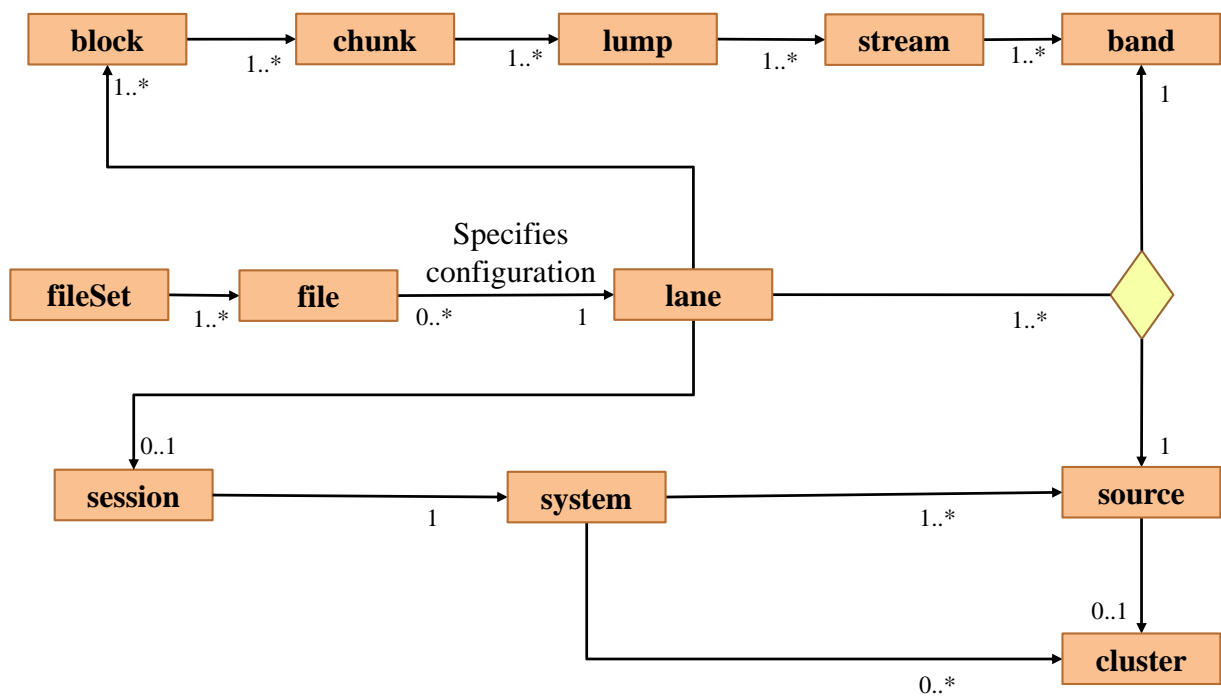
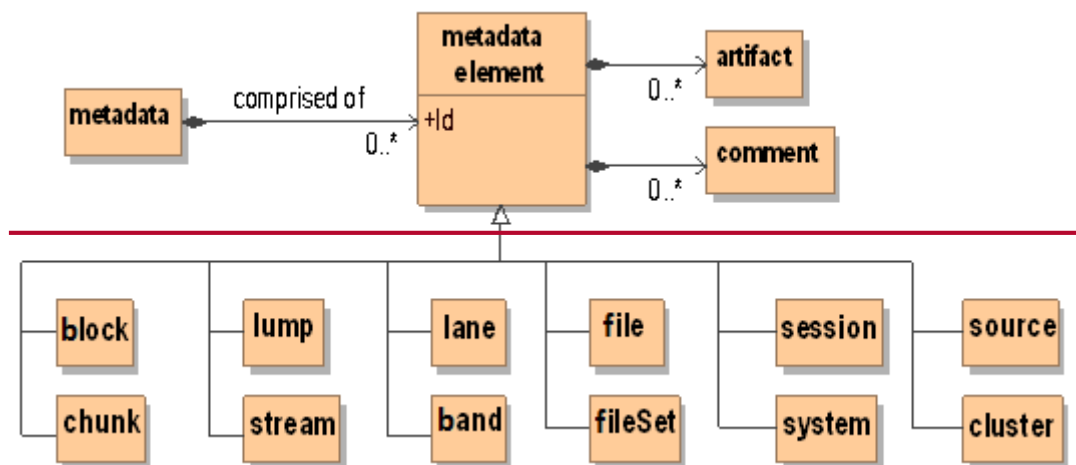


Figure 3 – GNSS metadata class model (UML 2.0)



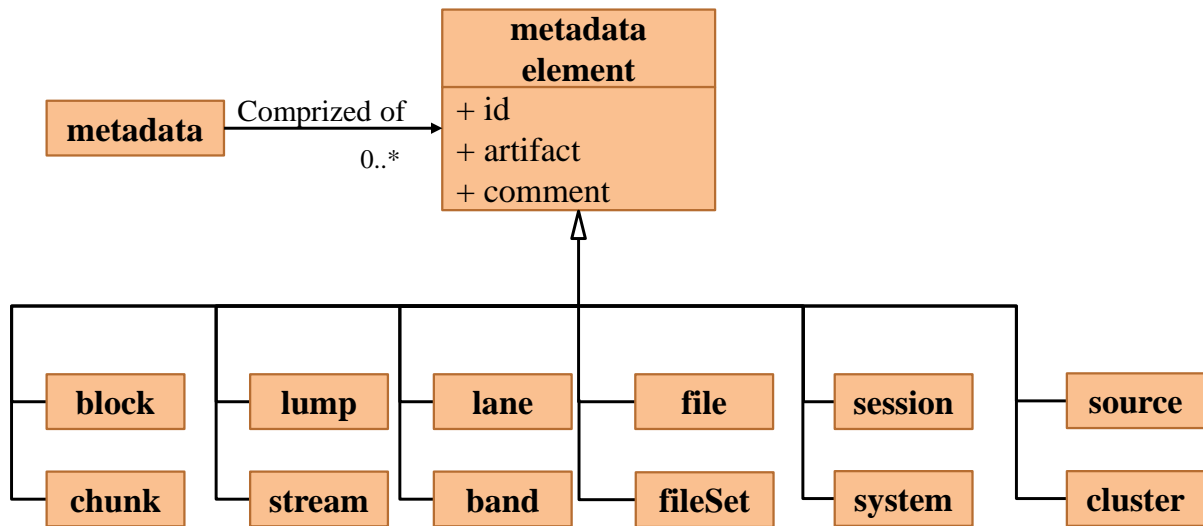


Figure 4 — Core metadata classes specialize the base metadata element, which has a unique identifier (id), links to related artifacts (URI) and comments

All metadata objects contain the following attributes:

- **artifact**: ~~One~~one or more generic attributes
- **comment**: one or more comment strings
- **id**: an identification string that is used to reference a child object by the parent

~~Table 1~~ Table 1 describes the attributes of the metadata element class. Core metadata classes specialize the base metadata element. It encapsulates a unique identifier (id), links to related artifacts (URI) and comment strings.

Table 1 Metadata element class attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>id</b>	Unique identifier	string		Yes	<del>""</del>
<b>artifact</b>	Zero or more link specifications to information pertaining to the class instance. Can be any URI formatted information	URI		Yes	
<b>comment</b>	Zero or more text/html comments providing additional detail regarding the class instance.	string		Yes	

## 5.2.6.2 Core classesClasses

### 5.2.16.2.1 Session object

A session is defined as a utilization instance of a pre-configured system for a period devoted to a particular activity.

Table 2 Definition of session attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>toa</b>	Time of applicability for all position and attitude parameters	dateTime <sup>1</sup>		No	
<b>position</b>	Platform position at toa expressed in <del>Geoid</del> ellipsoid frame	position		No	
<b>system</b>	The system used for this session	system		No	
<b>poc</b>	Point of contact. Name of <del>the</del> person or entity.	string		No	
<b>contact</b>	<del>POC</del> poc contact information (email)	string		No	
<b>Campaigncampaign</b>	Data collection campaign	string		No	
<b>Scenarioscenario</b>	Specific scenario for this collection	string		No	

<sup>1</sup> [https://www.w3schools.com/xml/schema\\_dtypes\\_date.asp](https://www.w3schools.com/xml/schema_dtypes_date.asp)

<sup>1</sup> [https://www.w3schools.com/xml/schema\\_dtypes\\_date.asp](https://www.w3schools.com/xml/schema_dtypes_date.asp)

### 5.2.26.2.2 System object

A system is defined as a complete data collection apparatus. The system comprises all antennas, sensors, and other information-outputting equipment down to the disk arrays that store SDR files. The system may also include GNSS signal simulators. The standard includes geometrical parameters (position and orientation) to the extent that this information is necessary for post-processing the SDR data stream. For example, initial position and platform orientation may be needed for a dynamic scenario. The relative position and orientation of antennas and their elements with respect to the platform coordinate frame are needed for adaptive antenna signal processing.

Table 3 — Definition of system attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
source	One or more sources of sampled data.	source		No	
cluster	Zero or more clusters of antenna sources	cluster		No	
fregbase	Base frequency. All sampling frequencies are specified as an integer multiple of fregbase	frequency		Yes	
equipment	Equipment used for this data collection	string		No	

### 5.2.36.2.3 Cluster object

Data collection setups may contain one or more antenna units where each antenna unit may comprise one or more elements. The position and orientation of each element's phase center and the relative delay must be known in order to perform multi-element signal processing. Hence, it is convenient to include these parameters directly as metadata. The standard defines the generic terms *cluster* and *source* to refer to an antenna unit and its elements respectively.

A cluster is defined as a grouping of sources. A coordinate frame is associated with a cluster. The origin and orientation of this frame is specified with respect to the platform coordinate frame.

Table 4 — Definition of cluster attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>id</b>	Unique identifier	string		No	
<b>position</b>	Origin of cluster reference frame w.r.t. platform coordinate frame	position		No	
<b>orientation</b>	Orientation of cluster frame w.r.t. platform frame	orientation		No	
<b>vendor</b>	Vendor name	string		No	
<b>model</b>	Model number	string		No	
<b>serial</b>	Serial number	string		No	

#### 5.2.46.2.4 Source Object

A source is defined as the originator of an electrical signal. A coordinate frame is associated with a cluster. The origin and rotation of this frame is specified with respect to the platform coordinate frame.

Table 5 Definition of source attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>id</b>	<del>cluster</del> Cluster that this source belongs to	string		No	
<b>type</b>	Electrical type of this source	<del>enum</del> string	"UndefinedType", "Patch", "Dipole", "Helical", "Quadrifilar", "Simulator", "Other"	No	"UndefinedType"
<b>polarization</b>	Element polarization	<del>enum</del> string	"UndefinedType", "RHCP", "LHCP", "Linear", "Horizontal", "Vertical"	No	"UndefinedType"
<b>origin</b>	Origin with respect to <del>the</del> cluster	position		No	
<b>orientation</b>	Orientation of normal vector to this source plane w.r.t. cluster	orientation		No	

### 5.2.56.2.5 Band object

A band is defined as a span of RF spectrum. Each band is received from a single source and converted to a sample stream by a signal processor that is typically referred to as an RF front-end. This analog signal represented by the band experiences the following changes as it passes through this mixed-signal processing chain:

- The RF center frequency,  $F_{RF}$ , is translated to  $F_{IF}$ .
- The spectrum may become inverted such that the frequency  $F_{RF}+dF$  is translated to  $F_{IF}-dF$ , where  $dF$  is a frequency offset from  $F_{RF}$ .
- The sampled representation of the band is delayed with respect to the signal incident at the phase center of the source (i.e. antenna element). This delay may vary with time, and is hence defined at the system time of applicability,  $t_{oa}$ .
- An approximate double-sided half power bandwidth can be specified for the stream representation of the band.

The above are specified in terms of band attributes.

Table 6 – Definition of band attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>centerfreq</b>	Center frequency of band incident at source	frequency		Yes	
<b>translatedfreq</b>	Translated center frequency of band	frequency		Yes	
<b>inverted</b>	Binary flag indicating spectral inversion	boolean	“true”, “false”	No	“false”
<b>delaybias</b>	Delay of band measured from source to sampled stream, specified at toa	duration		No	0
<b>bandwidth<sup>1</sup></b>	Approximate double-sided half power bandwidth	frequency		No	

<sup>1</sup> Bandwidth is measured by processing the sample stream. For streams containing multiple bands, it is recommended that other bands be muted to measure a given bandwidth.

#### 5.2.66.2.6 Stream object

A frequency-translated signal may contain more than one band. For example, in a direct RF sampling front-end, the sample rate may be chosen such that multiple passbands are intentionally aliased to fall adjacent to one another in the spectrum of the sampled signal. This is illustrated in Figure 5.

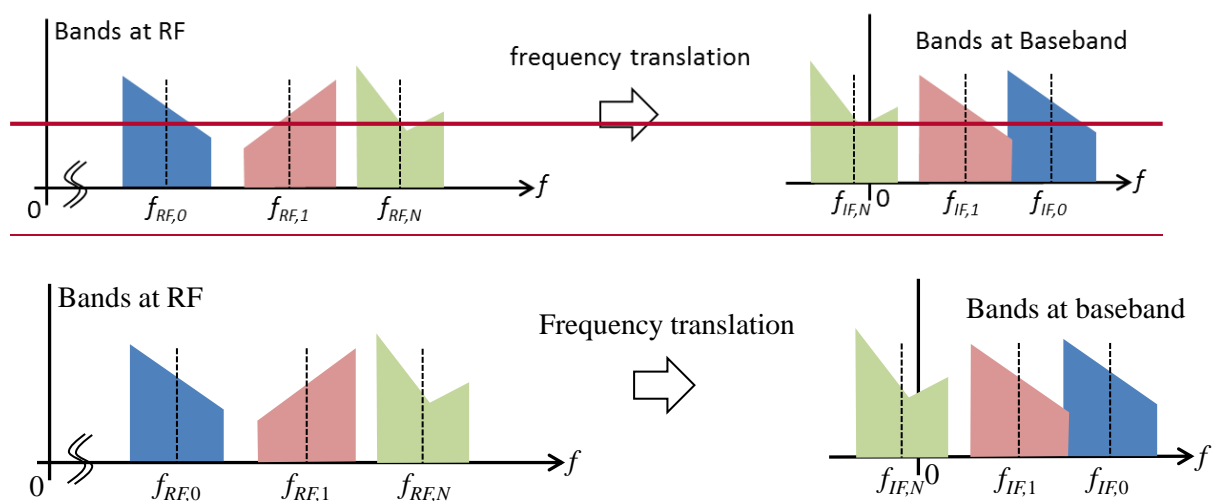


Figure 5 – Intentional Aliasing of a Multiband signal to Baseband

Figure 6 illustrates the conceptual representation of the digitization of a signal containing multiple bands. The output of this process is a sampled representation of the multi-band signal referred to as a sample stream.



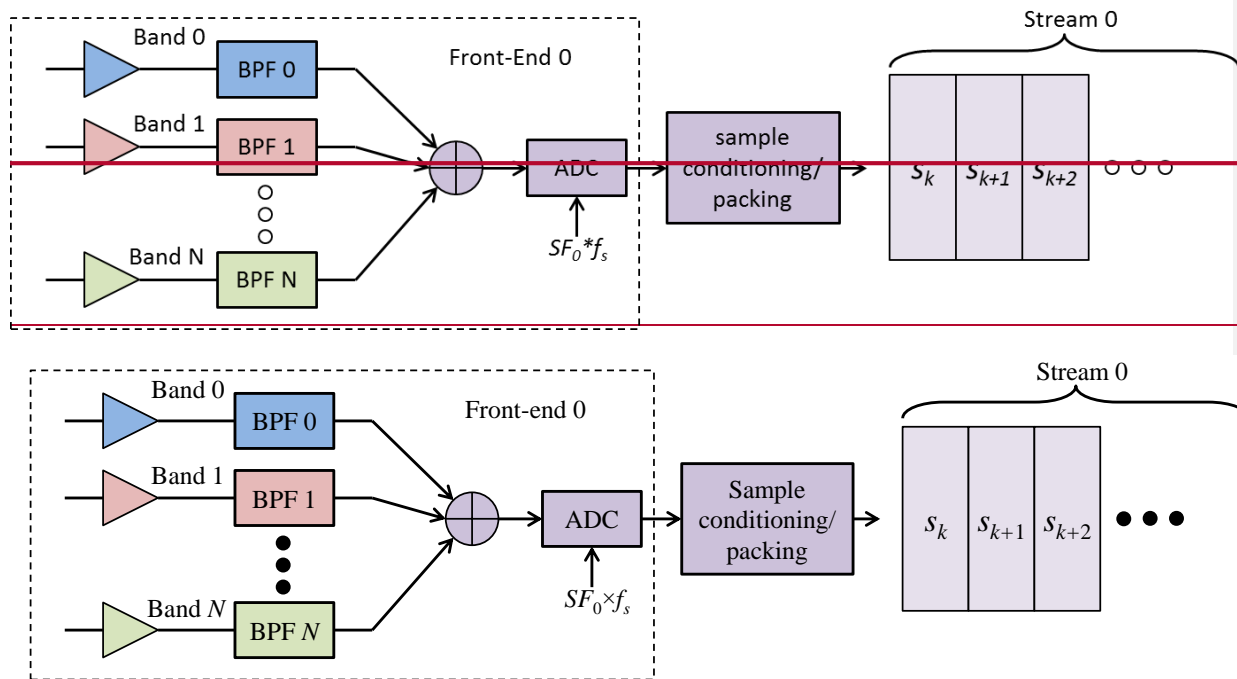


Figure 6 – Illustration of ~~Multiple Bands Present~~ multiple bands present in a ~~Stream~~ stream

Formatted  
(States)

Formatted  
(States)

Formatted  
(States)

Formatted  
(States)

Formatted

Formatted

Formatted

Formatted

Formatted

Formatted

A (sample) stream is defined as a discrete-time discrete-amplitude series that is the sampled representation of a combination of one or more bands.

A stream has the following properties:

- The stream contains the sampled representation of one or more bands.
- A stream is sampled at a given sample rate. This sample rate may be different to other streams in the system. The sample rate of a stream is specified as an integer multiple (ratefactor) of the system base sample rate (freqbase). As such, freqbase should represent the highest common integer factor of the sample rates of all streams.
- Sample values may be real or complex depending on whether IF sampling or baseband sampling is used, respectively. Some or all the numerical values expressed in the stream may be inverted.
- Each sample value is represented by one or more bits which may be encoded using various established schemes. The value quantization should reflect the number of bits required to express all quantization levels, being rounded up when the number of quantization levels is not a power of two (i.e. three-level quantization requires two bits).
- The value packedbits represents the total number of bits occupied by the collection of samples contained in a chunk (the chunk is a segment of data packed in one of the unsigned integer standards, a more accurate detailed description of the chunk is given in section 6.2.8) in the stream where:  

$$\text{packedbits} \geq \text{ratefactor} \times \text{quantization},$$
for real data, and:  

$$\text{packedbits} \geq 2 \times \text{ratefactor} \times \text{quantization},$$
for complex data.
- When the above inequality holds, the alignment of the quantized samples with respect to the packed samples must be known in order to interpret the sample values correctly.

The above are specified in terms of stream attributes.

Table 7 — Definition of stream attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>band<sup>1</sup></b>	One or more bands present in this stream	band		Yes	
<b>ratefactor</b>	Sample rate factor	uint16_t		Yes	
<b>quantization</b>	Sample quantization (bitsbit)	uint8_t		Yes	
<b>packedbits</b>	Packed representation (bitsbit)	uint8_t		Yes	
<b>alignment</b>	Sample alignment	enum #string	"Left" "Right" "Undefined"	Yes	
<b>shift</b>	Shift direction	enum #string	"Left" "Right" "Undefined"	Yes	

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>format</b>	Sample representation	<del>enum</del> <del>string</del>	<del>"IF"</del> <del>"IFn"</del> <del>"IQ"</del> <del>"IQn"</del> <del>"InQ"</del> <del>"InQn"</del> <del>"QI"</del> <del>"QIn"</del> <del>"QnI"</del> <del>"QnIn"</del> <del>"(where 'n' signifies means inversion)"</del>	Yes	
<b>encoding</b>	Numeric encoding scheme	<del>enum</del> <del>string</del>	<del>See Table 8</del> <del>Table 8 enumerates the different stream encoding formats supported in this standard.</del> <del>Table 8 enumerates the different stream encoding formats supported in this standard.</del>	Yes	

<sup>1</sup> Multiple instances of these parameters may exist. The parser shall enumerate accordingly.

Table 8 enumerates the different stream encoding formats supported in this standard.

Table 8 — Enumeration of stream encoding attribute

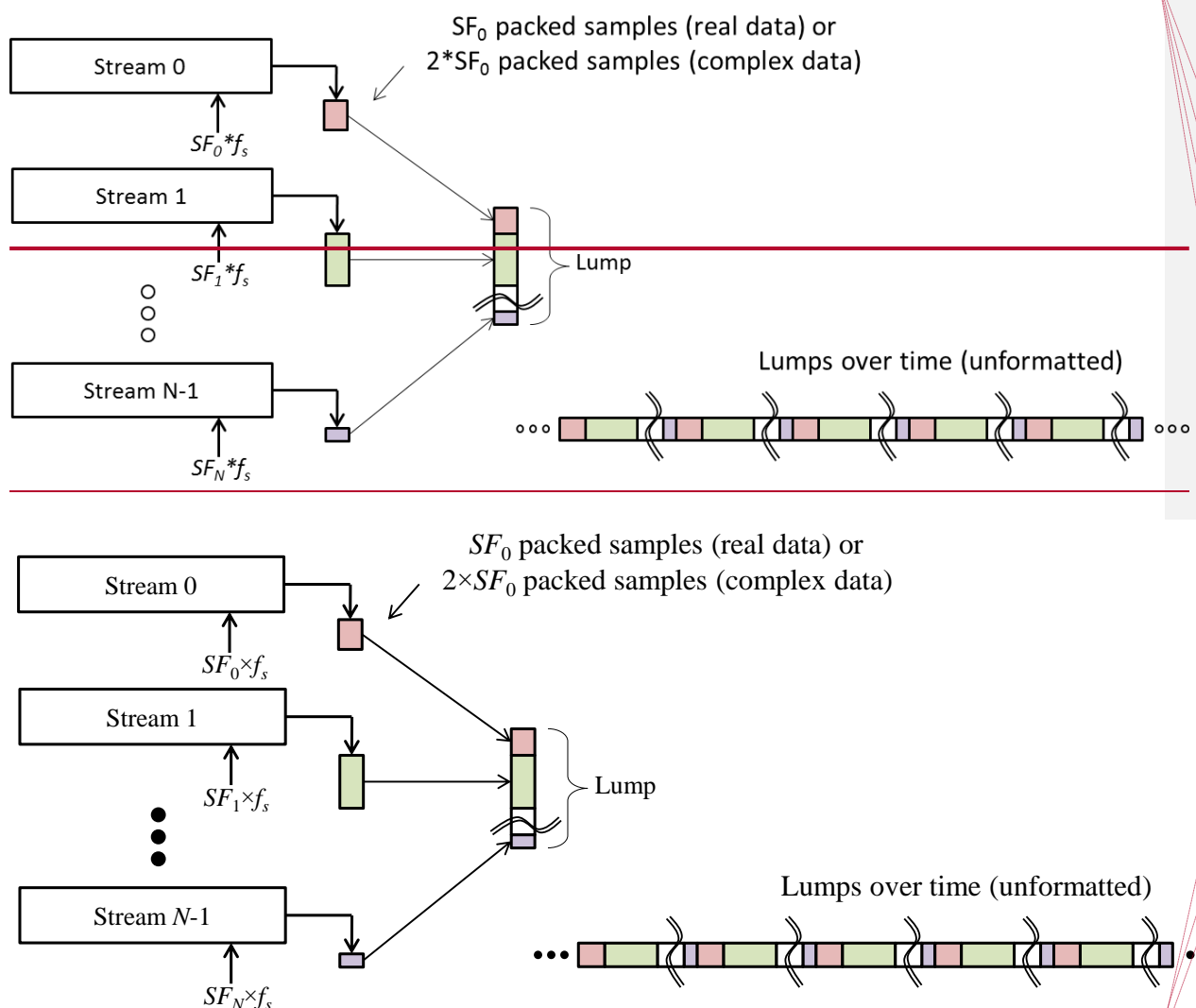
XML String	Description
"SIGN"	Sign
"OB"	Offset-Binary
"SM"	Sign-Magnitude
"MS"	Magnitude-Sign
"TC"	Two's <del>Compliment</del> Complement
"OG"	Offset-Gray Code
"OBA"	Offset-Binary Adjusted
"SMA"	Sign-Magnitude Adjusted
"MSA"	Magnitude-Sign Adjusted
"TCA"	Two's <del>Compliment</del> Complement Adjusted
"OGA"	Offset-Gray Code Adjusted

### 5.2.76.2.7 Lump object

Samples from two or more sample streams may be time multiplexed to form a single data stream that is ultimately written to disk (after additional formatting is applied, as described later in this document). This standard assumes that all samples belonging to a finite interval of time are packed into a contiguous grouping of bits, known as a **lump**.

A lump is defined as the ordered containment of all samples occurring within an interval  $t_s = 1/f_s$ . As more than one sample from each stream may exist within a given lump, the variable *shift* indicates which sample is chronologically first. When *shift* is set to “Left” the samples located at the most significant bits are the earliest, and when it is set to “Right” the samples located at the least significant bits are the earliest.

**Figure 7** Figure 7 illustrates a lump containing all samples from  $N$  sample streams.



**Figure 7** Illustration of a lump containing samples from  $N$  streams

Table 9 — Definition of lump attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>stream</b>	One or more streams present in this lump (ordered)	stream		No	
<b>shift</b>	Shift direction	enum	“Left”, “Right”	No	

### 5.2.86.2.8 Chunk object

The packing scheme of samples in a data stream must be known to correctly decode them. For example, consider 32 1-bit real samples packed into two uint16\_t words represented in little-endian format. Due to the little-endian representation, these samples will be decoded incorrectly if read back as a single uint32\_t word and shifted out. Further, some systems pack samples from left to right within a word whereas others perform the opposite.

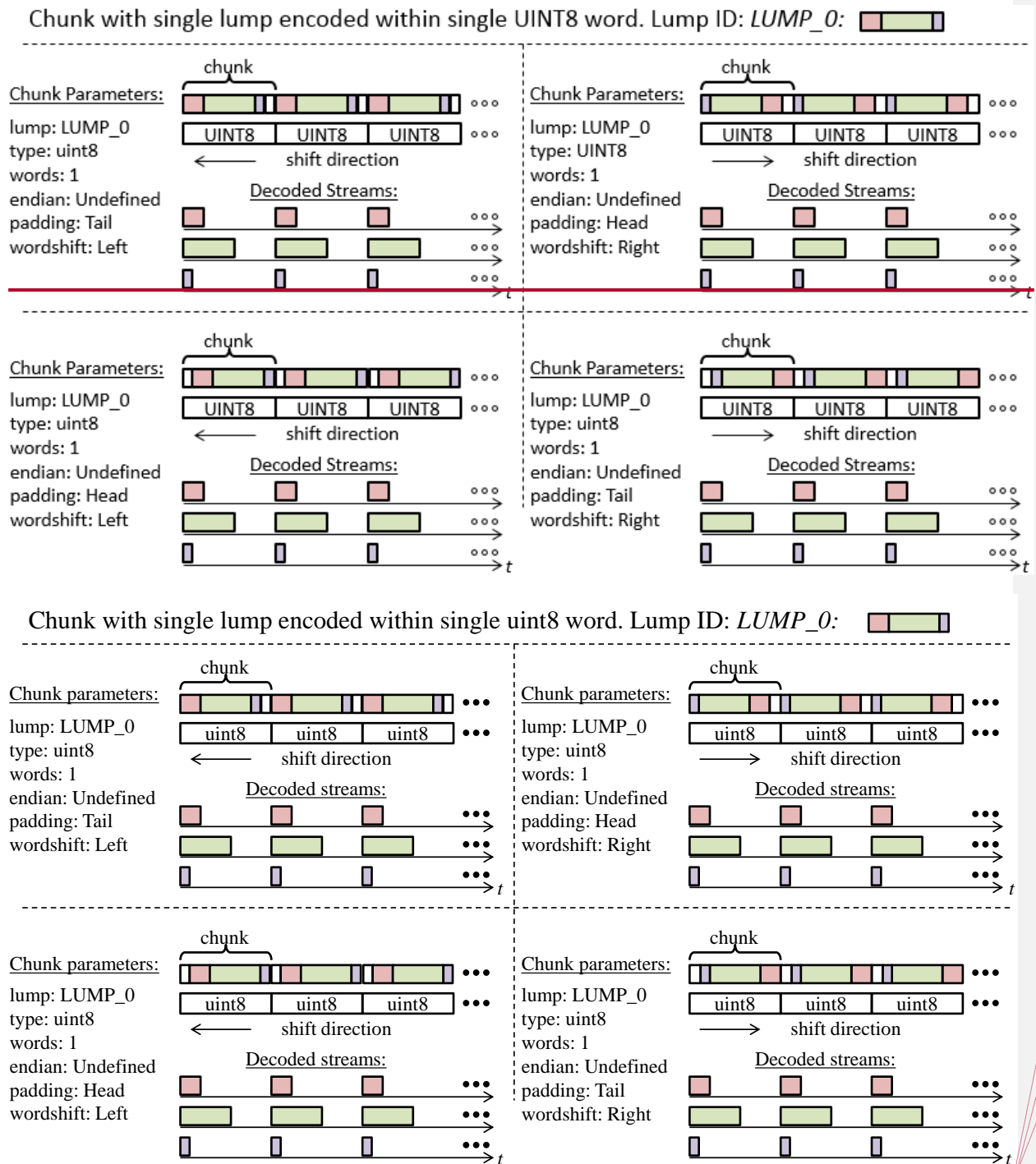
This standard defines a metadata parameter known as a chunk that together with stream and lump parameters unambiguously describes how samples shall be decoded from a data stream.

A chunk is defined as a segment of data consisting of one or more lumps that have been packed using one of four standard unsigned integer data types (uint8\_t, uint16\_t, uint32\_t, or uint64\_t). This provides a means of describing the occupied memory in a manner that can be natively manipulated by a processor, using standard memory structures (char, int, array).

Table 10-11 - Definition of chunk attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>lump</b>	One or more lumps	lump		Yes	
<b>sizeword</b>	The size, in bytes, of the fundamental integer data-type (word) that shall be read	uint8_t	1, 2, 4, 8 (Corresponds to uint8_t, uint16_t, uint32_t and uint64_t)	Yes	
<b>countwords</b>	Total number of words to be read in order to read/decode this chunk	uint8_t		Yes	
<b>endian</b>	Endianness of words stored in a chunk	enum string	"Big", "Little", "Undefined"	No	"Little"
<b>padding</b>	Padding applied during encoding	enum string	"None", "Head", "Tail"	No	"None"
<b>wordshift</b>	Shift direction	enum string	"Left", "Right"	Yes	

**Figure 8** Figure 8 illustrates four different schemes where a single 7-bit lump may be encoded within a chunk. The number of bits of information contained within a lump (and hence the number of bits to discard while decoding a chunk – shown as whitespace) is determined implicitly by parsing the referenced lump and stream parameters.



**Figure 8** Encoding schemes for a single lump within a single chunk

### 5.2.96.2.9 Block object

A data stream may contain other undefined bytes of information. This standard includes parameters necessary to skip over these bytes while decoding sample streams. This information is contained within a metadata object referred to as a block.

A block has the following properties:

- A block is comprised of a finite integer number of chunks greater than zero.
- Chunks within a block are sequential and contiguous.
- A block may begin with a data segment of arbitrary size (integer number of bytes) known as a *header*.
- A block may end with a data segment of arbitrary size (integer number of bytes) known as a *footer*.
- A block may contain data integrity features that are implemented within the header and/or footer segments.
- The block data structure shall remain constant for the entire data collection session (i.e. block format shall not change dynamically).



A block is defined as a data segment comprised of one or more chunks, where the chunk data appears contiguously anywhere within said segment.

Table 11-13 - Definition of block attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
chunk	One or more chunks	chunk		Yes	
cycles	For the ordered chunk pattern described in the attribute chunk, the integer number of cycles that this pattern repeats within a block	uint32_t		Yes	
Sizeheader, sizeheader	Integer number of bytes to skip in order to access first byte of chunk data	uint32_t		No	0
sizefooter	Integer number of bytes to skip in order to access first byte of next block	uint32_t		No	0

Figure 9 illustrates the encoding of  $N$  chunks within a block with no header and footer, with header only, with footer only, and with both data segments.

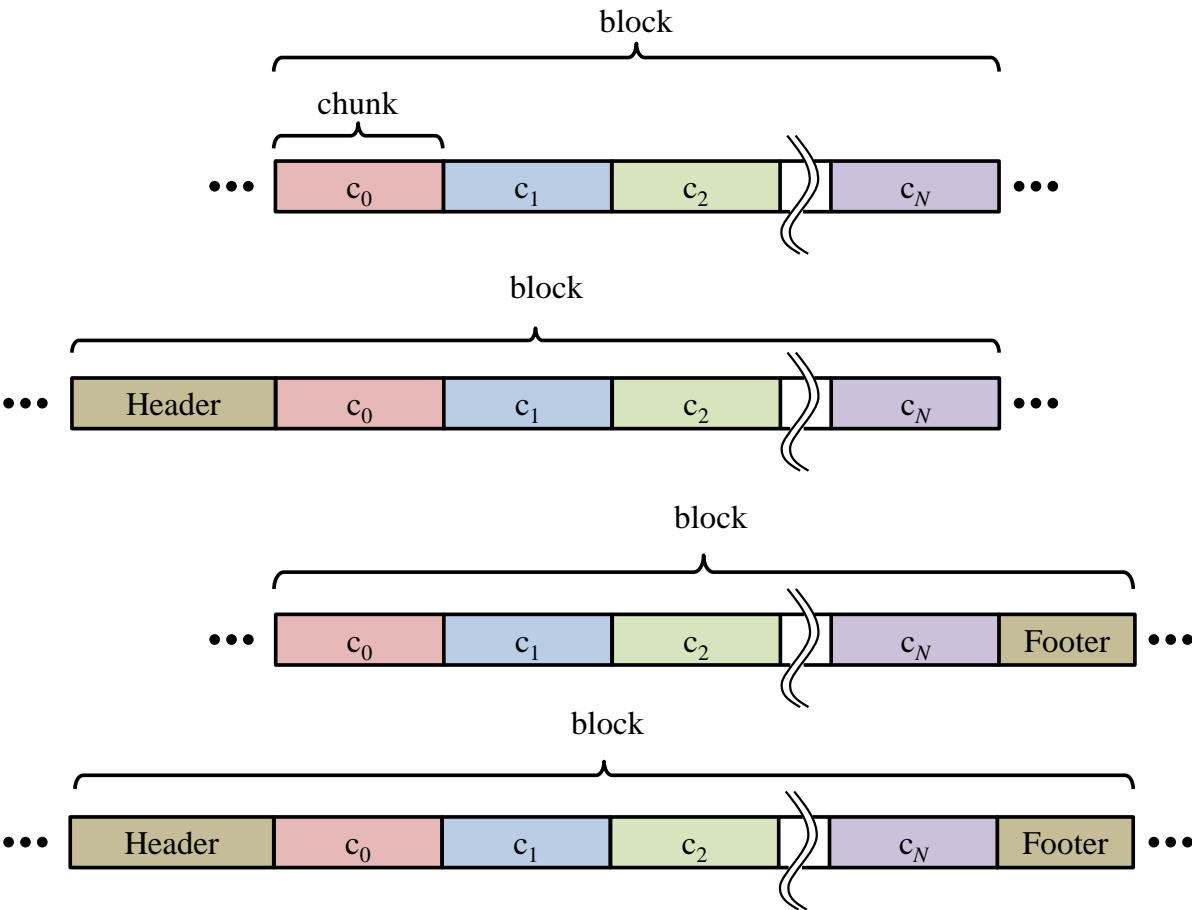


Figure 9 - Encoding schemes for  $N$  chunks within a block

### 5.2.106.2.10 Lane object

A lane is defined as a conduit that transports data comprised of one or more types of blocks. The contents of one or more lanes are written to disk to produce files. However, the standard does not assume that this writing is synchronized to the start of a block within a lane.

Table 12 Definition of lane attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>block</b>	One or <b>more</b> types of blocks in this lane (in order)	block		Yes	
<b>bandsrc</b>	Associates predefined bands with sources	string		Yes	
<b>session</b>	Session information for this lane	session		Yes	
<b>system</b>	System information for this lane	system		Yes	

Figure 10 illustrates  $N$  blocks contiguously encoded within a lane.

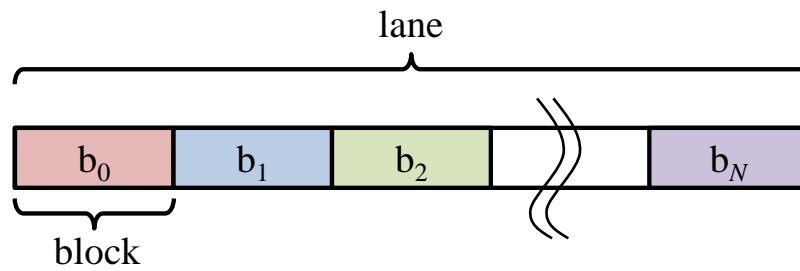


Figure 10 - Encoding scheme for  $N$  blocks within a lane

### 5.2.11 File object

A file is defined as the ordered collection of bytes retrieved from a single lane over a finite interval of time and stored in a digital media device.

When a lane is written to a file, it may or may not be synchronized to the start of a block. For this reason there may be a byte offset from the beginning of the file to the first byte of the first block. This offset may be different for each file.

The creation time of the file may be tagged as metadata. This time is typically obtained from the system RTC.

Table 13-16 - Definition of file attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>url</b>	Unique identifier for the file (path/filename)	URI		Yes	
<b>timestamp</b>	Time the file was generated	dateTime		No	
<b>offset</b>	Byte offset to start of first block	uint32_t		No	0
<b>lane</b>	Identifies which lane the data came from	lane		Yes	
<b>previous</b>	Name of previous file (for temporally split files)	URI		No	
<b>next</b>	Name of next file (for temporally split files)	URI		No	
<b>owner</b>	String specifying the owner of this file	string		No	
<b>copyright</b>	Copyright information	string		No	

Figure 11 illustrates the encoding of  $N$  lanes within a file.

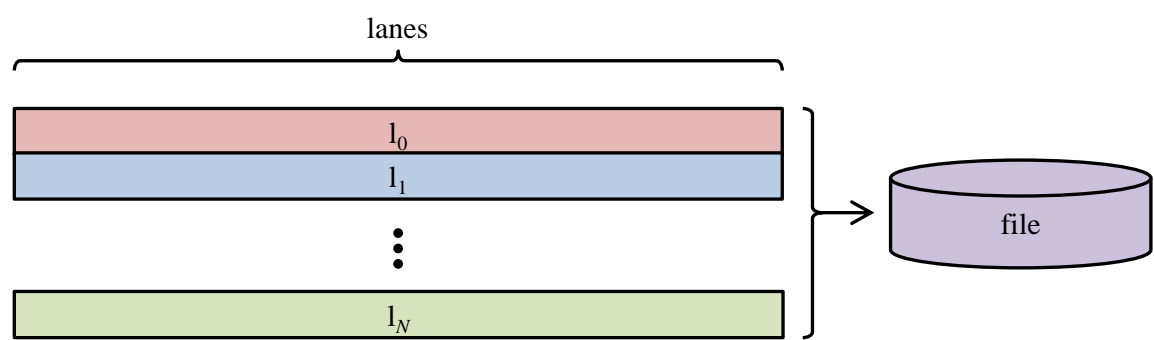


Figure 11 - Encoding of the lanes within a file

5.2.126.2.12 FileSet Object

For spatially and spatial-temporally split files, the file set must be identified. This is done by the FileSet parameters that identify the *first set of files*. All other information can be obtained by parsing the metadata of those files.

Table 14 - Definition of fileSet attributes

Attribute	Description	Class	Enumeration	RequiresR <u>equied</u>	Default (if no specified)
file	Names of files comprising the file set	URI		Yes, for spatial or spatial- temporal	

### 5.3.6.3 Foundation Classes

The domain object model foundation classes define basic types used by the core metadata elements.

#### 5.3.6.3.1 URI

A Universal Resource Identifier (URI) defines a unique path (e.g. URL) for locating an associated resource. The URI type is used to enable specification in a XML compatible format.

#### 5.3.6.3.2 DateTime

The dateTime string specifies the day of the year and the time in standard XML format. See [https://www.w3schools.com/xml/schema\\_dtypes\\_date.asp](https://www.w3schools.com/xml/schema_dtypes_date.asp)

An example of the representation of this type of parameter is shown below, as a definition of the time of applicability (toa):

```
<toa>2014-12-30T22:38:54.905999999Z</toa>
```

In the example is possible to check the day of applicability (30<sup>th</sup> of December of 2014) and the time of applicability (22:38:54.905999999).

#### 5.3.6.3.3 Frequency

Specifies frequency. Units can be Hz, kHz, MHz, or GHz. **Format** The format can be double, exponential or a ratio of the form. The ratio format is represented as frequency = 'xxxx-/yyyy' where frequency = xxx-/yyyy where xxx and yyyy are signed and unsigned 32-bit integers, respectively.

An example of the representation of this type of parameter is shown below, as a definition of the centerfreq parameter:

```
<centerfreq format="Hz">1227600000e+000</centerfreq>
```

```
<centerfreq format="MHz">1227.600</centerfreq>
```

```
<centerfreq format="kHz"> 1227600/0000</centerfreq>
```

In the example is possible to see that the center frequency of the received signal is 1227600000Hz, 1227 600 000 Hz, or 1.2276GHz, 2276 GHz.

#### 5.3.6.3.4 Duration

Used for specifying an interval of time. Units include ns, us, ms, s, which are represented in the XML file in the following way: "ns", "us", "ms" and "sec". Format is double, and thus negative values are supported. An example of the representation of this parameter type is shown:

```
<delaybias format="sec">0.0000000000000000e+000</delaybias>
```

### 5.3.56.3.5 Position

The position attribute is used to specify the location of the platform with respect to the [Geoidellipsoid](#). For a dynamic scenario, this is typically the initial location.

Table 15 — Definition of position attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>datum</b>	Datum used for the <a href="#">Geoidellipsoid</a>	string	“WGS-84”	No	“WGS-84”
<b>lat</b>	The latitude coordinate of the position <a href="#">in degrees</a>	double	<a href="#">[-90, 90]</a>	Yes	
<b>lon</b>	The longitude coordinate of the position <a href="#">in degrees</a>	double	<a href="#">[-180, 180]</a>	Yes	
<b>height</b>	The height coordinate of the position <a href="#">in meters</a>	double		Yes	

An example of the position is shown:

```
<position lat="48.17154012" lon="11.80868949" height="576.860"/>
```

### 5.3.66.3.6 Origin

Represents the origin of a child reference frame with respect to the parent reference frame.

Table 16 — Definition of origin attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
<b>lat</b>	The latitude coordinate of the origin <a href="#">in degrees</a>	double	<a href="#">[-90, 90]</a>	Yes	
<b>lon</b>	The longitude coordinate of the origin <a href="#">in degrees</a>	double	<a href="#">[-180, 180]</a>	Yes	
<b>height</b>	The height coordinate of the origin <a href="#">in meters</a>	double		Yes	

### 5.3.76.3.7 Orientation

Orientation defines a rotation from a parent coordinate frame to a child frame (i.e. this frame). By default, the rotation is specified in terms of a  $[3 \times 1]$  set of Euler angles. Other forms are supported by enumerating the class attribute (if it exists).

Table 17 Definition of orientation attributes

Attribute	Description	Class	Enumeration	Required	Default (if not specified)
type	Type of rotation matrix used	enumeration	"Euler"	No	"Euler"
value	Rotation values, in degrees	double	[0, 360]	Yes	

### 6.3.8 String

The string class is defined by Unicode Transformation Format 8 (UTF-8). See [https://www.w3schools.com/charsets/ref\\_html\\_utf8.asp](https://www.w3schools.com/charsets/ref_html_utf8.asp)



## 6 Appendix

### I: Encoding Functions

Below are examples of each of the sample encoding schemes which can be specified in the Stream attributed 'encoding' for a selection of bit widths including 2, 3, 4, and 5-bit digitization. The first column, entitled 'Binary' represents the binary data packed in the stream, MSB left, while the remaining columns represent the physical amplitude of the sample.

Table 18 Encoding of 2-bit samples

Binary	OB	OBA	SM	SMA	MS	MSA	TC	TCA	OG	OGA
00	-2	-3	0	1	0	1	0	1	-2	-3
01	-1	-1	1	3	0	-1	1	3	-1	-1
10	0	1	0	-1	1	3	-2	-3	1	3
11	1	3	-1	-3	-1	-3	-1	-1	0	1

Table 19 Encoding of 3-bit samples

Binary	OB	OBA	SM	SMA	MS	MSA	TC	TCA	OG	OGA
000	-4	-7	0	1	0	1	0	1	-4	-7
001	-3	-5	1	3	0	-1	1	3	-3	-5
010	-2	-3	2	5	1	3	2	5	-1	-1
011	-1	-1	3	7	-1	-3	3	7	-2	-3
100	0	1	0	-1	0	5	-4	-7	3	7
101	1	3	-1	-3	0	-5	-3	-5	2	5
110	2	5	-2	-5	1	7	-2	-3	0	1
111	3	7	-3	-7	-1	-7	-1	-1	1	3

Table 20 Encoding of 4-bit samples

Binary	OB	OBA	SM	SMA	MS	MSA	TC	TCA	OG	OGA
0000	-8	-15	0	1	0	1	0	1	-8	-15
0001	-7	-13	1	3	0	-1	1	3	-7	-13
0010	-6	-11	2	5	1	3	2	5	-5	-9
0011	-5	-9	3	7	-1	-3	3	7	-6	-11
0100	-4	-7	4	9	0	5	4	9	-1	-1
0101	-3	-5	5	11	0	-5	5	11	-2	-3
0110	-2	-3	6	13	1	7	6	13	-4	-7
0111	-1	-1	7	15	-1	-7	7	15	-3	-5
1000	0	1	0	-1	0	9	-8	-15	7	15
1001	1	3	-1	-3	0	-9	-7	-13	6	13
1010	2	5	-2	-5	1	11	-6	-11	4	9
1011	3	7	-3	-7	-1	-11	-5	-9	5	11
1100	4	9	-4	-9	0	13	-4	-7	0	1
1101	5	11	-5	-11	0	-13	-3	-5	1	3
1110	6	13	-6	-13	1	15	-2	-3	3	7
1111	7	15	-7	-15	-1	-15	-1	-1	2	5

Table 21 Encoding of 5-bit samples

Binary	OB	OBA	SM	SMA	MS	MSA	TC	TCA	OG	OGA
00000	-16	-31	0	1	0	1	0	1	-16	-31
00001	-15	-29	1	3	0	-1	1	3	-15	-29
00010	-14	-27	2	5	1	3	2	5	-13	-25
00011	-13	-25	3	7	-1	-3	3	7	-14	-27
00100	-12	-23	4	9	0	5	4	9	-9	-17
00101	-11	-21	5	11	0	-5	5	11	-10	-19
00110	-10	-19	6	13	1	7	6	13	-12	-23
00111	-9	-17	7	15	-1	-7	7	15	-11	-21
01000	-8	-15	8	17	0	9	8	17	-1	-1
01001	-7	-13	9	19	0	-9	9	19	-2	-3
01010	-6	-11	10	21	1	11	10	21	-4	-7
01011	-5	-9	11	23	-1	-11	11	23	-3	-5
01100	-4	-7	12	25	0	13	12	25	-8	-15
01101	-3	-5	13	27	0	-13	13	27	-7	-13
01110	-2	-3	14	29	1	15	14	29	-5	-9
01111	-1	-1	15	31	-1	-15	15	31	-6	-11
10000	0	1	0	-1	0	17	-16	-31	15	31
10001	1	3	-1	-3	0	-17	-15	-29	14	29
10010	2	5	-2	-5	1	19	-14	-27	12	25
10011	3	7	-3	-7	-1	-19	-13	-25	13	27
10100	4	9	-4	-9	0	21	-12	-23	8	17
10101	5	11	-5	-11	0	-21	-11	-21	9	19
10110	6	13	-6	-13	1	23	-10	-19	11	23
10111	7	15	-7	-15	-1	-23	-9	-17	10	21
11000	8	17	-8	-17	0	25	-8	-15	0	1
11001	9	19	-9	-19	0	-25	-7	-13	1	3
11010	10	21	-10	-21	1	27	-6	-11	3	7
11011	11	23	-11	-23	-1	-27	-5	-9	2	5
11100	12	25	-12	-25	0	29	-4	-7	7	15
11101	13	27	-13	-27	0	-29	-3	-5	6	13
11110	14	29	-14	-29	1	31	-2	-3	4	9
11111	15	31	-15	-31	-1	-31	-1	-1	5	11

## Appendix II: Future Extensions

In future extensions of the SDR sampled data metadata standard, new format types will be available for representing object or their attributes. This section will explain some of the future features that will be introduced in the standard.

Currently, the datum attribute, which belongs to the position object (in section 6.3.5), is only supported by the WGS-84 format used by Global Positioning System (GPS). In future extensions new formats will be added to the standard, such as Galileo Terrestrial Reference Frame (GTRF) used by Galileo, and Parametry Zemli 1990 (PZ-90) used by GLONASS.

The position is also limited to the use of “Latitude, longitude, height” (LLH) format. In future implementation the “earth-centered, earth-fixed” (ECEF) format will also be available to represent this object. This will also be implemented in the object origin (in section 6.3.6).

Finally, more information will also be added to define the receiver, such as the initial platform velocity and the sample rate frequency error. The latter will be represented in parts per million (ppm).