

International Telecommunication Union

ITU-R
Radiocommunication Sector of ITU

Recommendation ITU-R BS.2125-0
(01/2019)

**A serial representation of the Audio
Definition Model**

BS Series
Broadcasting service (sound)

Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from <http://www.itu.int/ITU-R/go/patents/en> where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

Series of ITU-R Recommendations

(Also available online at <http://www.itu.int/publ/R-REC/en>)

Series	Title
BO	Satellite delivery
BR	Recording for production, archival and play-out; film for television
BS	Broadcasting service (sound)
BT	Broadcasting service (television)
F	Fixed service
M	Mobile, radiodetermination, amateur and related satellite services
P	Radiowave propagation
RA	Radio astronomy
RS	Remote sensing systems
S	Fixed-satellite service
SA	Space applications and meteorology
SF	Frequency sharing and coordination between fixed-satellite and fixed service systems
SM	Spectrum management
SNG	Satellite news gathering
TF	Time signals and frequency standards emissions
V	Vocabulary and related subjects

Note: This ITU-R Recommendation was approved in English under the procedure detailed in Resolution ITU-R 1.

Electronic Publication
Geneva, 2020

© ITU 2020

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without written permission of ITU.

RECOMMENDATION ITU-R BS.2125-0*

A serial representation of the Audio Definition Model

(2019)

Scope

This Recommendation describes a format of metadata based on the Audio Definition Model (ADM) specified in Recommendation ITU-R BS.2076, segmented into a time-series of frames. The XML format is used for the serial presentation of the ADM as is the case with the original ADM. The serial presentation of the ADM is designed for use in linear workflows such as live or real-time production for broadcasting and streaming applications. This Recommendation does not cover the transport method or bit-packing of the metadata, or the format of the audio samples to which the metadata relates.

Keywords

ADM, Audio Definition Model, Serialization, Segmentation, Advanced sound system, Multichannel audio, Channel-based, Object-based, Scene-based, Metadata, Immersive audio

The ITU Radiocommunication Assembly,

considering

- a) that storage media of all types based on Information Technology have penetrated all areas of audio production for radio broadcasting, namely non-linear editing, on-air play-out and archives;
- b) that when programme material is either produced live or distributed as live, real-time streaming over communication and broadcast networks is required;
- c) that the adoption of a single streaming metadata format for live interchange would greatly simplify the interoperation of individual pieces of equipment, and of remote studios;
- d) that compatibility with current audio-related metadata, the Audio Definition Model (ADM), specified in Recommendation ITU-R BS.2076, would minimize the efforts required for format conversion;
- e) that for linear workflows such as live and real-time production for broadcast and streaming, frame-based or serialized audio and metadata is needed;
- f) that future audio systems will require metadata associated with the audio to be carried in linear streams;
- g) that advanced sound systems will use a variety of configurations including channel-, object-, and scene-based audio such as specified in Recommendation ITU-R BS.2051;
- h) that advanced sound systems will use the Audio Definition Model (ADM) specified in Recommendation ITU-R BS.2076 to describe the technical format of the audio being delivered and exchanged;
- i) that advanced sound systems will use the BW64 audio file format specified in Recommendation ITU-R BS.2088 to exchange audio programmes with the ADM,

* Radiocommunication Study Group 6 made editorial amendments to this Recommendation in February 2020 in accordance with Resolution ITU-R 1.

recommends

- 1 that, for workflows requiring serialized metadata based on the Audio Definition Model (ADM), the serial representation of the ADM described in Annex 1 should be used;
- 2 that Note 1 be considered part of the Recommendation.

NOTE 1 – Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words “shall” or some other obligatory language such as “must” and the negative equivalents are used to express requirements. The use of such words shall in no way be construed to imply partial or total compliance with this Recommendation.

Annex 1

A serial representation of the Audio Definition Model (ADM)

A1.1 Introduction

The advanced sound system specified in Recommendation ITU-R BS.2051 requires audio-related metadata to handle the channel-based, object-based and scene-based audio elements. Recommendation ITU-R BS.2076 provides the Audio Definition Model (ADM) described in Extensible Markup Language (XML) as the audio-related metadata for advanced sound systems.

Recommendation ITU-R BS.2088 provides the 64-bit audio file format of “BW64”, which can store XML code of the ADM metadata in the ‘axml’ chunk and is used to exchange audio programmes of advanced sound systems.

However, the ADM is not suitable for live production and streaming audio applications. Live production and streaming audio applications involve either slicing an existing audio file into frames, or generating frames, and delivering these frames in real-time over delivery interfaces (such as AES3 (Recommendation ITU-R BS.647), MADI (Recommendation ITU-R BS.1873), HD-SDI (Recommendations ITU-R BT.1120 and BT.1365) and IP networks). Therefore, a serial format for the ADM is required to allow slicing of audio and its associated metadata.

This Recommendation describes how the ADM can be represented in a serial metadata format for use in live production and streaming audio applications of advanced sound systems. It covers the segmentation of metadata and the serial format of the metadata. This serial format has the following features:

- It is compatible with the structure, attributes and elements of the ADM.
- It has no limitation on the number of audio tracks that may be described.
- It is independent of transport or interface method.
- It can handle any combination of channel-, object-, and scene-based audio programmes as described by Recommendation BS.2076.
- There is no restriction on frame size.
- It allows for random-access support.

This Recommendation does not describe how to carry, constrain or transport the Serial ADM (S-ADM) frames in specific interfaces, nor does it describe how to associate the S-ADM metadata frames with the audio essence.

A1.1.1 Definitions

Audio Essence – The data of the audio signal(s), either represented in samples or coded data representing those samples.

Chunk – Is a subset of metadata elements from an S-ADM frame.

Flow – A sequence of S-ADM frames is called a flow. A flow is the S-ADM equivalent of what a file would be in normal ADM. Therefore, a flow may contain one or more **audioProgrammes**. The flow is identified with a **flowID** in the form of a UUID.

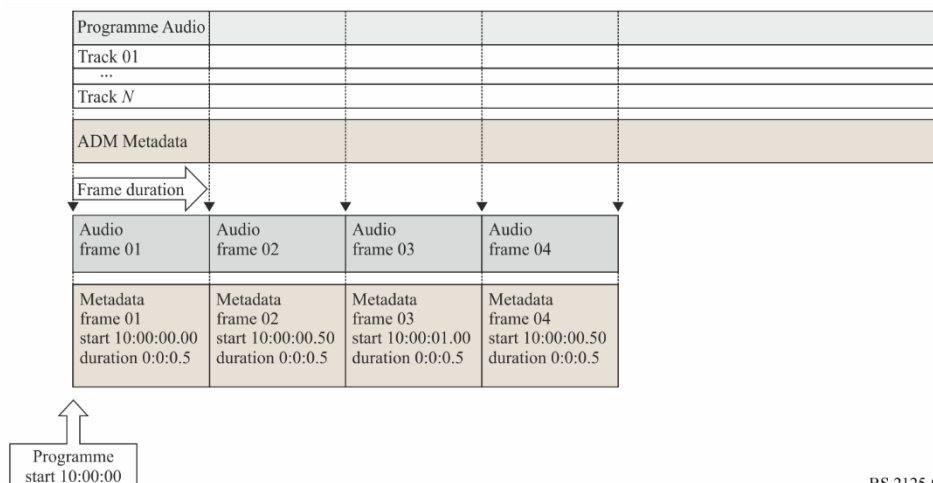
Random-Access – The ability to access any frame in a flow and fully decode it. In the context of this Recommendation it is the ability to access an arbitrary frame in a flow and extract all metadata needed for the corresponding audio frame. In some cases, more than one frame may be required to extract this metadata (random-access with a delay). When random access is not supported, all prior frames in the flow (starting with the first frame) may be required to extract metadata needed for a given audio frame.

A1.2 Overview of S-ADM

A frame of S-ADM metadata contains a set of metadata to describe at least the audio frame over the time period associated with that frame. S-ADM has the same structure, attributes and elements as those of ADM, as well as additional attributes to specify the frame format (see § A1.4). The S-ADM frames are non-overlapping and contiguous with a specified duration and start time. However, the metadata contained in an S-ADM frame may describe audio beyond the duration of that frame. The segmentation and transport of the audio frames is not covered in this Recommendation.

The ADM metadata consists of the content part, for instance, the **audioProgramme**, and the format part, for instance, the **audioChannelFormat**. Only the three elements **audioProgramme**, **audioObject** and **audioBlockFormat** have time-related parameters. In the content part, the **start**, **end** and **duration** of the **audioProgramme** or the **audioObject** determine the start time, end time or duration of the elements. These parameters are usually fixed. In the format part, all parameters in the **audioBlockFormat** are time-variant parameters.

FIGURE 1
Overview of the S-ADM



The ADM metadata can be separated into two groups, time-variant dynamic metadata (e.g. the **audioBlockFormat** in the **audioChannelFormat**) and time-invariant static metadata (e.g. **audioProgramme** and **audioContent**).

An S-ADM metadata frame shall consist of one or multiple metadata chunks.

S-ADM metadata frames are categorized into five types:

- ‘header’: indicates the first frame in a flow that has all descriptors associated with audio signals
- ‘full’: all descriptors associated with audio signals
- ‘divided’: metadata divided into chunks, with the very last chunk containing dynamic metadata and other chunks containing parts of the static metadata
- ‘intermediate’: only descriptors changed from the previous frame
- ‘all’: all descriptors for total of **audioProgramme** (whole XML codes of the original ADM)

An S-ADM flow shall be one of the following:

- Full-Frame (FF) flow: A series of ‘full’ frames, with the very first frame either ‘full’, ‘header’, or ‘all’.
- Intermediate-Frame (IF) flow: A series of ‘intermediate’ frames, with the very first frame either ‘full’, ‘header’, or ‘all’.
- Mixed-Frame (MF) flow: A series of ‘intermediate’ and ‘full’ frames, with the very first frame either ‘full’, ‘header’, or ‘all’.
- Divided-Frame (DF) flow: A series of ‘divided’ frames, with the very first frame either ‘full’, ‘divided’, ‘header’, or ‘all’.

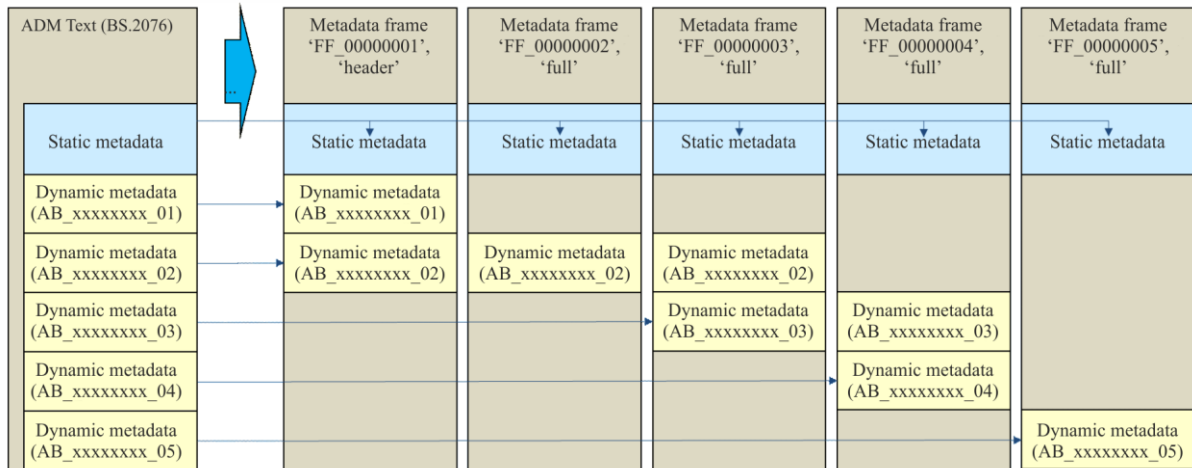
The purpose of the ‘divided’ and ‘intermediate’ frames is to allow efficient representations of the S-ADM data by not repeating time-invariant metadata in every frame. The S-ADM flow types are designed to support such efficient representations while providing random-access functionality when needed. The envisioned application of the S-ADM flows is as follows:

Use case	Recommended flow
<ul style="list-style-type: none"> – Metadata predominantly dynamic and changing every frame, or – Random-access is required for every frame 	Full-Frame (FF)
<ul style="list-style-type: none"> – Flow includes static or slowly-changing dynamic metadata, and – Random-access is NOT required 	Intermediate-Frame (IF)
<ul style="list-style-type: none"> – Flow includes static or slowly-changing dynamic metadata, and – Random-access is required (but not for every frame) 	Mixed-Frame (MF)
<ul style="list-style-type: none"> – Flow includes static or slowly-changing dynamic metadata, and – Random-access is required (not for every frame), and – It is desired to distribute data more evenly across all frames 	Divided-Frame (DF)

A1.2.1 Full-Frame (FF) flow explanation

In this case, the basic structure of the S-ADM is constructed into ‘full’ frames (shown in Fig. 1). The FF flow provides access to any audio frame for random-access support (see Fig. 2).

FIGURE 2
Basic structure of the S-ADM in the Full Frame (FF) flow

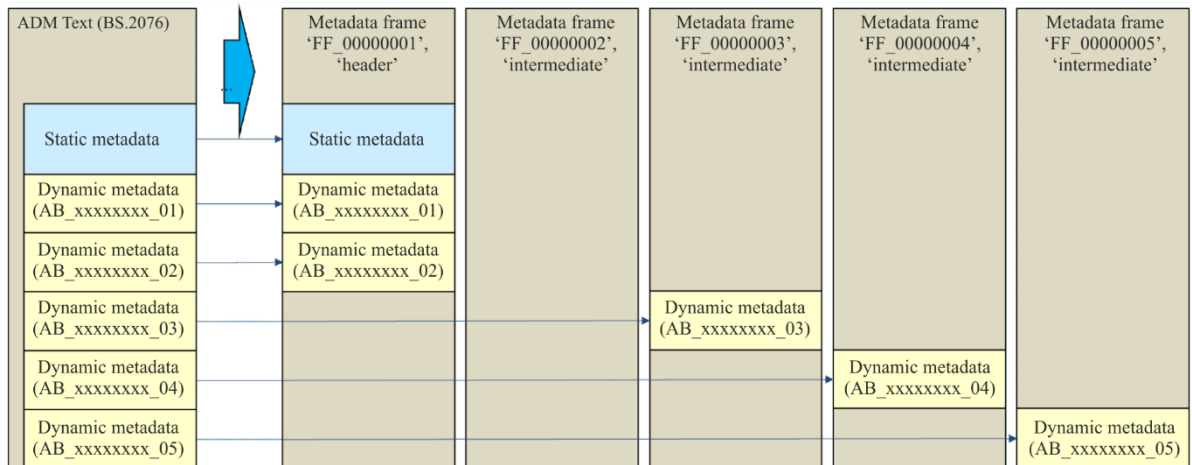


BS.2125-02

A1.2.2 Intermediate-Frame (IF) flow explanation

The receiver may only need to receive the static ADM metadata just once, so any repeated static ADM metadata can be ignored even if the full metadata is repeatedly transported. Therefore, when a broadcaster does not require random-access, ADM metadata that has been already transported can be omitted. The ‘intermediate’ frame can omit all elements whose values are unchanged from those in the previous frame even if the element is classified in the dynamic metadata. The IF flow does not support random-access (see Fig. 3).

FIGURE 3
Structure of the S-ADM in the Intermediate Frame (IF) flow



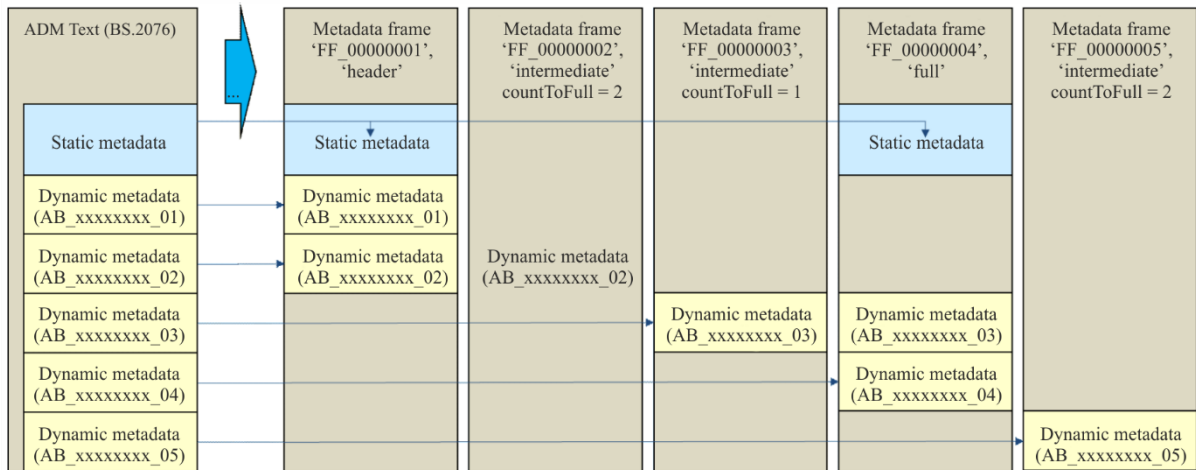
BS.2125-03

A1.2.3 Mixed-Frame (MF) flow explanation

Both ‘full’ and ‘intermediate’ frames can be used in the single stream (Fig. 4). In this case, a broadcaster freely determines intervals to transport the ‘full’ frames. The MF flow supports random-access with a delay: the receiver has to wait for the next ‘full’ frame.

FIGURE 4

Structure of the S-ADM in the Mixed Frame (MF) flow



BS.2125-04

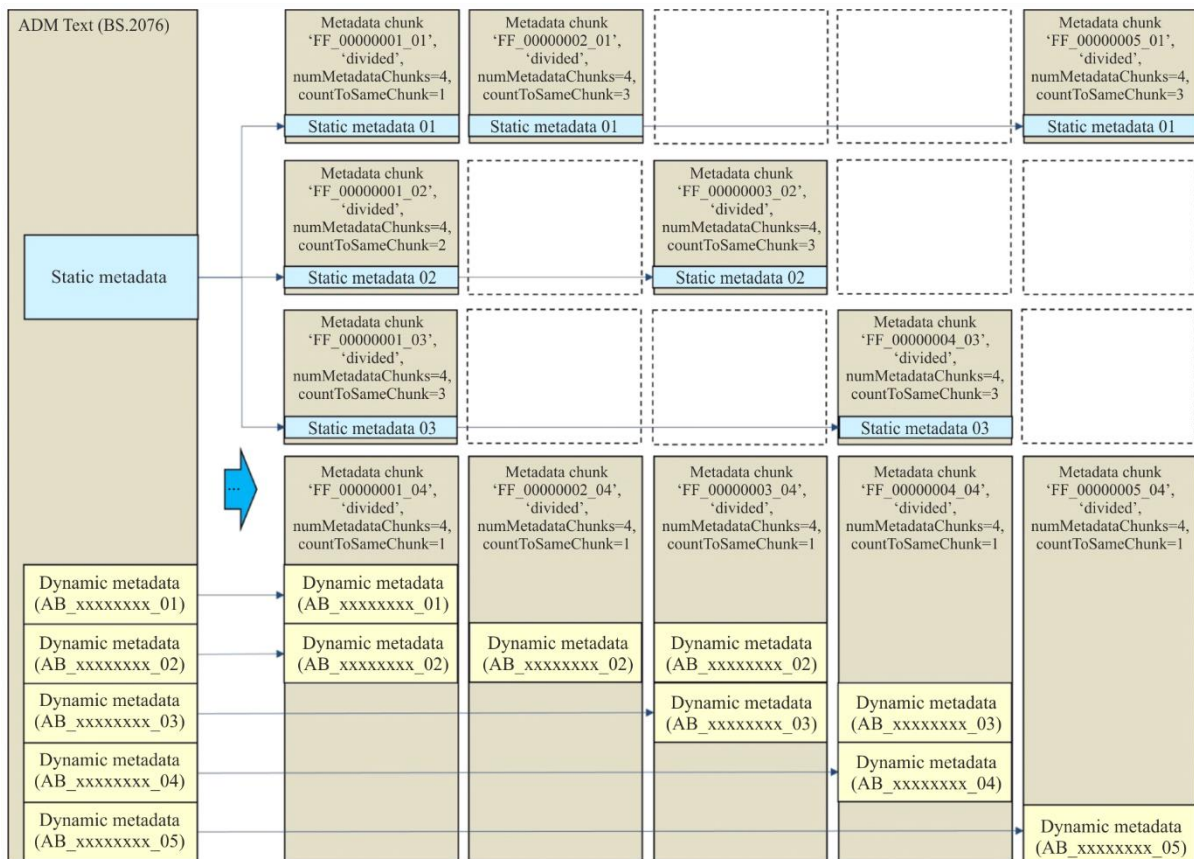
A1.2.4 Divided-Frame (DF) flow explanation

The MF flow has very changeable traffic depending on whether it is the 'full' frame or the 'intermediate' frame. With the static metadata divided into chunks, the DF flow is designed to distribute the data more evenly across all frames (Fig. 5).

Example from Figure 5: The frame metadata, e.g. 'FF_00000001', is divided into metadata chunks such as 'FF_00000001_01', 'FF_00000001_02' and 'FF_00000001_03'. These metadata chunks are transported at the same time instance. The metadata chunk 'FF_0000000X_04' includes dynamic metadata and the metadata chunks 'FF_0000000X_01' to 'FF_0000000X_03' include divided static metadata. Since the metadata chunk 'FF_00000002_01' has the same static metadata as other frames (e.g. 'FF_00000003_01' and 'FF_00000004_01'), therefore 'FF_00000003_01' and 'FF_00000004_01' can be omitted.

In the DF flow, the very last chunk always contains dynamic metadata, while all other chunks contain static metadata. The DF flow supports random-access with a delay: the receiver has to wait until all the metadata chunks needed to reconstruct the complete static metadata set are received.

FIGURE 5
Structure of the S-ADM in the Divided-Frame (DF) flow



BS.2125-05

A1.2.5 Real-time S-ADM generation

Some examples of how to generate S-ADM in a real-time environment are illustrated in Figs 6 to 8. The examples are shown for MF and FF flows, but similar procedures can be applied to the other flow types.

Figure 6 shows how the initialisation of an **audioObject** ('AO_1001') and some **audioBlockFormats** ('AB_00030001_NN') can occur in a real-time scenario. The duration of 'AO_1001' starts off as 2 seconds (to match the length of the frame) when it first appears in 'FF_00000003'; and then the **duration** is updated to 4 seconds, and then 6 seconds in the following frames. New **audioBlockFormats** appear in 'FF_00000003', 'FF_00000004' and 'FF_00000005', with some of their **duration** values adjusted when the block **audioBlockFormat** is used in the frame following the one it first appeared in.

The reconstructed ADM on the right of the diagram shows how the elements will appear after 'FF_00000005' has been received, so 'AO_1001' has a **duration** of 6 seconds.

FIGURE 6

Structure of the S-ADM in a real-time scenario illustrating how new elements are introduced and modified

Metadata frame 'FF_00000001', 'header' start: 10:00:00.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000002', 'intermediate' start: 10:00:02.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000003', 'Full' start: 10:00:04.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000004', 'intermediate' start: 10:00:06.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000005', 'Full' start: 10:00:08.00000 duration: 00:00:02.00000	Reconstructed ADM
'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000		'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000		'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000
		'AO_1001' start: 00:00:04.00000 duration: 00:00:02.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:04.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:06.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:06.00000
		'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0		'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0
			'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30		'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30
			'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:01.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60
				'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:01.00000 position Azimuth: 0

BS.2125-06

Figure 7 shows how a new **audioObject** is introduced, but does not contain any child elements for the first two frames. Therefore, its **start** time is modified on successive frames until it is allocated some child elements. In this case, when it gets to 'FF_00000003' a new **audioBlockFormat** appears ('AB_00030001_01'), so the 'AO_1001' **start** time of 4 seconds is fixed, and its **duration** increases through the subsequent frames.

FIGURE 7

Structure of the S-ADM in a real-time scenario illustrating how new elements without children is handled

Metadata frame 'FF_00000001', 'header' start: 10:00:00.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000002', 'intermediate' start: 10:00:02.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000003', 'Full' start: 10:00:04.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000004', 'intermediate' start: 10:00:06.00000 duration: 00:00:02.00000	Metadata frame 'FF_00000005', 'Full' start: 10:00:08.00000 duration: 00:00:02.00000	Reconstructed ADM
'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000		'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000		'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000
	'AO_1001' start: 00:00:04.00000 duration: 00:00:02.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:02.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:04.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:06.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:06.00000
		'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0		'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0
			'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30		'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30
			'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:01.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60
				'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:01.00000 position Azimuth: 0

BS.2125-07

Figure 8 shows how the **end** time of the **audioProgramme** ('APR_1001') is modified when a new frame ('FF_00000006') appears that occurs after the end of the original **end** time of 'APR_1001'. The **durations** of the 'AO_1001' and 'AB_00030001_04' are also modified in this new frame. The reconstructed ADM's 'APR_1001' **end** time is also updated as a result.

FIGURE 8

Structure of the S-ADM in a real-time scenario illustrating how existing elements are modified

Metadata frame 'FF_00000001', 'header'	Metadata frame 'FF_00000002', 'full'	Metadata frame 'FF_00000003', 'full'	Metadata frame 'FF_00000004', 'full'	Metadata frame 'FF_00000005', 'Full'	Metadata frame 'FF_00000006', 'full'	Reconstructed ADM
start: 10:00:00.00000 duration: 00:00:02.00000	start: 10:00:02.00000 duration: 00:00:02.00000	start: 10:00:04.00000 duration: 00:00:02.00000	start: 10:00:06.00000 duration: 00:00:02.00000	start: 10:00:08.00000 duration: 00:00:02.00000	start: 10:00:10.00000 duration: 00:00:02.00000	
'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:10.00000	'APR_1001' start: 10:00:00.00000 end: 10:00:12.00000	'APR_1001' start: 10:00:00.00000 end: 00:00:12.00000
		'AO_1001' start: 00:00:04.00000 duration: 00:00:02.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:04.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:06.00000	'AO_1001' start: 00:00:04.00000 duration: 00:00:08.00000	'AO_1001' start: 00:00:04.00000 duration: 10:00:08.00000
		'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0			'AB_00031001_01' rtime: 00:00:01.00000 duration: 00:00:01.00000 position Azimuth: 0
			'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30			'AB_00031001_02' rtime: 00:00:02.00000 duration: 00:00:01.00000 position Azimuth: 30
			'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:01.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60	'AB_00031001_03' rtime: 00:00:03.00000 duration: 00:00:02.00000 position Azimuth: 60
				'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:01.00000 position Azimuth: 0	'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:03.00000 position Azimuth: 0	'AB_00031001_04' rtime: 00:00:05.00000 duration: 00:00:03.00000 position Azimuth: 0

BS.2125-08

When reading an S-ADM frame where the properties of a particular metadata element has changed with respect to that of previous frames, then the metadata element of the most recent frame shall be used.

A1.3 S-ADM metadata frame structure

The S-ADM metadata frame shall consist of two parts. The first is **frameHeader** that contains additional elements for the S-ADM to describe specification of the ADM metadata frame, and the second is **audioFormatExtended** that contains the ADM metadata specified in Recommendation ITU-R BS.2076.

A1.3.1 Structure of 'full' frame

The 'full' frame shall contain all elements within the audioFormatExtended part.

A1.3.2 Structure of 'intermediate' frame

The 'intermediate' frame should include only elements that have changed values when compared with the previous ADM metadata frame. The ADM elements of the **audioProgramme**, the **audioObject** and the **audioBlockFormat** specify time information. The **audioBlockFormat** for the **typeDefinition** of 'DirectSpeakers' usually has time-invariant metadata, whereas the **audioBlockFormat** for the **typeDefinition** of 'Object' often has time-variant metadata. The 'intermediate' frame usually consists of the **audioBlockFormat** in the **audioChannelFormat** for the **typeDefinition** of 'Object'.

A1.3.3 Structure of the 'divided' frame

The 'divided' frame contains metadata segmented into at least two chunks. Each frame shall carry at least one chunk. Each chunk shall contain a subset of all the elements that would be carried in a complete frame. As static metadata elements are not changing over successive frames, it is not necessary to place them in all frames. Dynamic metadata elements that may change every frame shall be carried in the last chunk carried within the frame.

A1.3.4 Structure of the 'header' frame

A 'header' frame is a 'full' frame with the special function that signals the start of a new **audioProgramme** or the start of a new flow.

A1.3.5 Structure of the ‘all’ frame

An ‘all’ frame should contain all the metadata for an entire **audioProgramme**. Therefore, this may include metadata that describes the audio in past and future frames, as well as the current one.

The ‘all’ type of frame should only be used where the metadata for the entire **audioProgramme** is known in advance of streaming the S-ADM frames. Therefore, it should only be considered for either pre-recorded programmes or live programmes with entirely static metadata.

A1.3.6 Universal properties for each frame

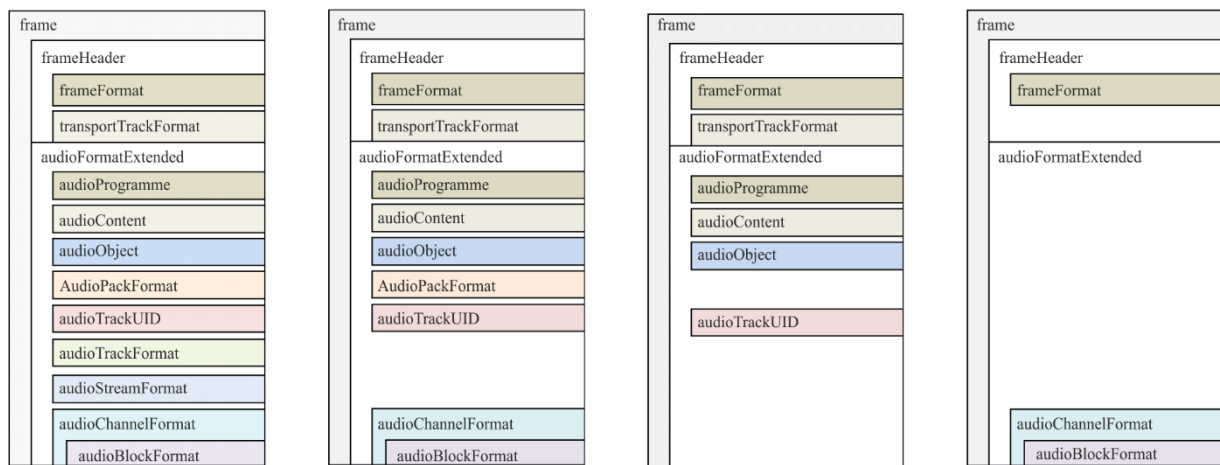
A1.3.6.1 PCM audio

For PCM audio, the **audioTrackFormat** shall refer to a single **audioStreamFormat** and the **audioStreamFormat** shall refer to a single **audioChannelFormat**. If the **audioTrackUID** directly refers to the **audioChannelFormat**, both the **audioTrackFormat** and the **audioStreamFormat** may be omitted.

A1.3.6.2 Common definitions

Recommendation ITU-R BS.2094 defines some typical use-cases of the ADM elements as the common definitions, especially for channel-based audio. When using the common definitions, the format part of the ADM including the **audioTrackFormat**, the **audioStreamFormat**, the **audioChannelFormat** and the **audioPackFormat** shall be omitted.

FIGURE 9
Structures of the ADM metadata in the frame for the S-ADM



BS.2125-09

A1.4 S-ADM elements and attributes

A1.4.1 frame, frameHeader and audioFormatExtended

The S-ADM parent element is the **frame**. It has two sub-elements the **frameHeader** and the **audioFormatExtended** sub-element. The **frameHeader** should be positioned first in each frame and it includes the **frameFormat** (see § A1.4.2) and the **transportTrackFormat** (see § A1.4.3), which are added to specify the structure of S-ADM metadata frame and to describe the transport audio interface for the S-ADM. The **audioFormatExtended** sub-element carries the ADM metadata as specified in Recommendation ITU-R BS.2076.

TABLE 1
frame sub-elements

Sub-Element	Description	Quantity
frameHeader	See § A1.4.1.1	1
audioFormatExtended	Contains the ADM metadata as described in Rec. BS.2076	1

A1.4.1.1 frameHeader

TABLE 2
frameHeader sub-elements

Sub-Element	Description	Quantity
frameFormat	Description of the format of the ADM metadata frame	1
transportTrackFormat	Description of the format of the transport audio interface	1...*

Example S-ADM XML is shown in § A2.1.

A1.4.2 frameFormat

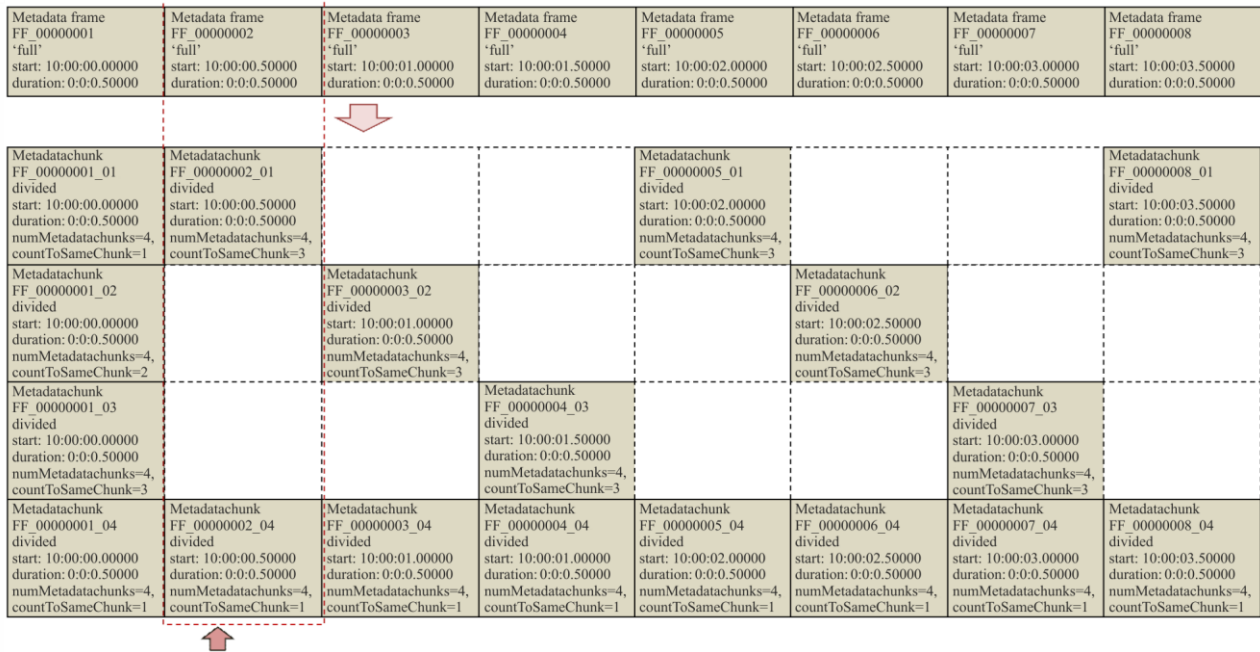
The **frameFormat** shall represent the specifications of the frame that contains the ADM elements and the specification of the associated audio frame. The **start** of the **frameFormat** indicates the elapsed time from the start time of the **audioProgramme**.

The synchronization and/or offset of the S-ADM metadata frames with the associated audio essence is left up to the transport/interface protocol.

The parameter **countToSameChunk** shall indicate the number of frames between the current frame and the frame where the same chunk reoccurs.

The parameter **numMetadataChunks** shall indicate the number of metadata chunks needed for random-access. The example in Fig. 10 has a **numMetadataChunks** value of 4. Note that the number of metadata chunks in each frame within a flow should be the same.

FIGURE 10
Use case of countToSameChunk



At least one metadata chunk should be used in the frame.

BS.2125-10

A1.4.2.1 Attributes

TABLE 3
frameFormat attributes

Attribute	Description	Example	Required
frameFormatID	ID of frame. ID is shown as format 'FF_XXXXXXXXXX_ZZ'. The part of 'FF_XXXXXXXXXX' is the number of the frames from start time. '_ZZ' is only used in the 'divided' frame where it shall indicate the index of the current metadata chunk starting with '_01'.	<ul style="list-style-type: none"> FF_00000000001 FF_00000000001_01 	Yes

TABLE 3 (*end*)

Attribute	Description	Example	Required
start	<p>Start time of frame. The following formats are available:</p> <ul style="list-style-type: none"> – ‘hh:mm:ss.zzzzz’ indicates hours, minutes and seconds of the elapsed time from the start time of the programme. – ‘yyyy-mm-ddThh:mm:ss.zzzzzZ’, where ‘yyyy-mm-dd’ indicates year, month and day. – ‘zzzzzSffff’, where ‘zzzzz’ is the number of audio samples elapsed from the start time of the programme and ‘ffff’ indicates the sampling rate. – ‘hh:mm:ss.zzzzzSffff’ indicates hours, minutes and seconds of the elapsed time from the start time of the programme, where ‘zzzzz’ is not time but audio samples and ‘ffff’ indicates the sampling rate. 	<ul style="list-style-type: none"> – 00:00:00.00000 – 1970-01-01T00:00:00.00000Z – 0S48000 – 09:59:59.47999S48000 – 2017-12-31T23:59:59.47999S48000 	Yes
duration	<p>Duration of frame. Formats available are:</p> <ul style="list-style-type: none"> – ‘hh:mm:ss.zzzzz’ – ‘hh:mm:ss.zzzzzSffff’ – ‘zzzzzSffff’ – ‘ss.zzzzz’ <p>Note: the number of ‘z’ and ‘f’ digits can be from 5 to 9 digits, depending upon the desired precision.</p>	<ul style="list-style-type: none"> – 00:00:00.25000 – 00:00:00.12000S48000 – 12000S48000 – 00.25000 	Yes
type	<p>Descriptor of the type of frame. see §§ 3.1 to 3.5</p>	<ul style="list-style-type: none"> – header – full – divided – intermediate – all 	Yes
timeReference	<p>Descriptor of the time mode for the audioBlockFormat timing parameters. ‘total’ indicates that elapsed time from the audioProgramme start time is used. ‘local’ indicates that elapsed time from the beginning of the frame is used. This parameter shall be fixed for an entire flow. See § A1.4.4 for more detail about audioBlockFormat.</p>	<ul style="list-style-type: none"> – total – local 	Optional (default: total)
flowID	<p>Unique ID for a S-ADM sequence of frames described by the UUID specified in RFC 4122 or ISO/IEC 11578:1196.</p>	12345678-abcd-4000-a000-112233445566	Optional

TABLE 4

frameFormat attributes for frame types of ‘header’, ‘full’, ‘intermediate’ and ‘all’

Attribute	Description	Example	Required
countToFull	In the FF flow: set to ‘1’. In the MF flow: number of frames to the next ‘full’ frame. In the IF flow: set to ‘0’.	– 0 – 1 – 2 – 3 ...	Optional (default: 1 in FF, 0 in IF)

TABLE 5

frameFormat attributes for frame type of ‘divided’

Attribute	Description	Example	Required
numMetadataChunks	Number of metadata chunks needed for random-access. The number of the metadata chunks shall be the same in each frame within a flow.	– 2 – 3 ...	Yes
countToSameChunk	Number of frames to the next reoccurrence of a given static metadata chunk. If set to ‘1’ then each frame contains the static metadata chunk.	– 1 – 2 – 3 ...	Optional (default: unknown)

Note on time format and decimal places

The time-based time formats in this document show five decimal places for the seconds (either ‘ss.zzzzz’ or ‘hh:mm:ss.zzzzz’), but this is a minimum number of decimal places. It is acceptable to use more decimal places, and this is advised when sampling rates over 48 kHz are being used. Nine decimal places (i.e., hh:mm:ss.zzzzzzzzz) gives nanosecond precision.

For the longer form sample-based time format (hh:mm:ss.zzzzzSffff), the ‘z’ digits indicate a number of samples, the number of ‘z’s must match the number of ‘f’s. (i.e. ‘hh:mm:ss.zzzzzS48000’, ‘hh:mm:ss.zzzzzS192000’). The value of ‘zzzzz’ should be less than the value of ‘ffff’.

For the shorter sample-based format ‘zzzzzSffff’, the number of digits may be variable (i.e. ‘0S48000’ or ‘500000S48000’). The value of ‘zzzzz’ can be more than the value of ‘ffff’ if the time being represented is greater than a second.

The optional attributes **countToFull** and **countToSameChunk** are useful for the receiver to know when it may be able to start random-access playback. However, these attributes are not required to support the random-access functionality: even if these attributes are not used, the receiver can identify from the received data when all metadata needed for random-access is received.

A1.4.2.2 Elements

The **changedIDs** element of the **frameFormat** can show the ADM elements whose values are changed from the previous frames.

TABLE 6
frameFormat elements

Element	Description	Example	Quantity
changedIDs	List of references to ADM element IDs that have changed from the previous frame. Contains sub-elements describes in Table 6.	See § A2.2 for example XML.	0...1
chunkAdmElement	Reference to an ADM element categorized in each metadata chunk, if frameFormatID has multiple metadata chunks.	audioChannelFormat	0...*

The **changedIDs** sub-element can clearly show only the ADM elements whose values are changed from the previous frames. The sub-elements for **changedIDs** are shown in Table 7.

TABLE 7
changedIDs sub-elements

Sub-element	Attribute	Description	Example	Quantity
audioChannelFormatIDRef	status	Reference to an ID of audioChannelFormat changed from the previous frame. The attribute 'status' indicates changed descriptor is added or modified.	AC_00031001	0...*
audioPackFormatIDRef	status	Reference to an ID of audioPackFormat changed from the previous frame.	AP_00031001	0...*
audioTrackUIDRef	status	Reference to an ID of audioTrackUID changed from the previous frame.	ATU_00000001	0...*
audioTrackFormatIDRef	status	Reference to an ID of audioTrackFormat changed from the previous frame.	AT_00031001_01	0...*
audioStreamFormatIDRef	status	Reference to an ID of audioStreamFormat changed from the previous frame.	AS_00031001	0...*
audioObjectIDRef	status	Reference to an ID of audioObject changed from the previous frame.	AO_1001	0...*
audioContentIDRef	status	Reference to an ID of audioContent changed from the previous frame.	ACO_1001	0...*
audioProgrammeIDRef	status	Reference to an ID of audioProgramme changed from the previous frame.	APR_1001	0...*

TABLE 8
Status attribute

Attribute	Description	Example	Required
status	The status attribute is used to indicate generation of a new element, a changed element, an extension of an element, or expiration of an element.	<ul style="list-style-type: none"> • ‘new’ • ‘changed’ • ‘extended’ • ‘expired’ 	Yes

The status attribute shall be one of four values:

- ‘new’ – shall be used when a new element appears for the first time;
- ‘changed’ – shall be used when any parameters or values change within an element since the previous frame;
- ‘extended’ – shall be used when the timing parameters change since the previous frame, but all other parameters remain the same;
- ‘expired’ – shall be used when an element no longer exists in the current frame, but appeared in the previous frame.

Section A2.2 shows some example XML code to illustrate the usage of **changedIDs**.

A1.4.3 transportTrackFormat

The **transportTrackFormat** represents the relationship between physical audio tracks (e.g. channel 1 of AES3 interface) and the **UIDs** of the audio tracks in the ADM (e.g. ‘ATU_0000001’). In case of the ADM, this information is described in the ‘chna’ chunk of the BW64 file. The **transportTrackFormat** is the S-ADM equivalent of a BW64 ‘chna’ chunk.

A1.4.3.1 Attributes

The **transportName** is the name of the interface used for the transport of the associated audio essence. The specific names of the interface are not defined in this Recommendation. The users can freely use any name for the interfaces. When multiple interfaces are used, the interfaces may be labeled as device-A, device-B and device-C. The **numTracks** is the number of associated audio tracks in each interface. The **numIDs** is the number of the associated **audioTrackUIDs** in each interface.

TABLE 9
transportTrackFormat attributes

Attribute	Description	Example	Required
transportID	Index of the interface to transport the audio signal	TP_0001	Yes
transportName	Descriptor of the interface to transport the audio signal	AES3-A, AES3-B, etc.	Optional
numTracks	Number of associated transport tracks in each interface	16	Optional
numIDs	Number of associated audioTrackUIDs in each interface	32	Optional

A1.4.3.2 Elements

The **trackID** of the **audioTrack** is the index of the transport audio track in each interface. This index is the equivalent to the audio track number in the BW64 file. The **formatLabel** and **formatDefinition** indicate the format type of the audio signal. The values of these are defined in Recommendation ITU-R BS.2076.

TABLE 10
transportTrackFormat elements

Element	Coordinate attribute	Description	Example	Required
audioTrack	trackID	Index of the transport track in each interface. For instance, it is set to 1 or 2 for an AES3 interface (Rec. ITU-R BS.647), 1 to 64 in a MADI interface (Rec. ITU-R BS.1873)	1	Yes
	formatLabel	Descriptor of the format of the audio samples. When the formatLabel and formatDefinition are omitted, the formatLabel is intended as '0001'	0001	Optional
	formatDefinition	Description of the format of the audio samples. When the formatLabel and formatDefinition are omitted, the formatLabel is intended as 'PCM'	PCM	Optional

Both the **audioTrackFormatIDRef** and the **audioPackFormatIDRef** are not included in the **transportTrackFormat** so they should be referred by the **audioTrackUID**. Both the **audioTrackFormat** and the **audioStreamFormat** can be omitted for the PCM audio and the **audioTrackUID** can directly refer to the **audioChannelFormat** instead of the **audioTrackFormat**. Then, the same number is used for the IDs of both the **audioTrackFormat** and the **audioChannelFormat**.

TABLE 11
audioTrack sub-element

Sub-element	Description	Example	Quantity
audioTrackUIDRef	Reference to an audioTrackUID in the ADM code	ATU_00000001	0...*

Example code to illustrate the use of **transportTrackFormat** is in § A2.3.

A1.4.4 audioBlockFormat

The **audioBlockFormat** is an existing ADM element, and this section describes additional attributes for the S-ADM on top of the existing ADM **audioBlockFormat** attributes.

If the **timeReference** (attribute in **frameFormat**) is set to 'local', the **lstart** and **lduration** elements are used instead of the **rtime** and **duration** elements in the **audioBlockFormat**. The **lstart** and **lduration** elements represent the start time and duration of the audio block relative to the start time of the S-ADM frame.

Time-variant parameters (such as **position**) in the **audioBlockFormat** that overlaps the current frame might be defined at times outside the frame of S-ADM. **Istart** and **Iduration** enable the inclusion of this information without recalculation. For this purpose, **Istart** can be negative (i.e. before the start of the frame), and/or **Istart+Iduration** can be beyond the end of the frame. If time-variant parameters need to be placed on the frame boundaries, then recalculation of the parameters may be required.

The time-variant parameters in **audioBlockFormat** define the values at the end of the block. The values at the start of the block are defined by the previous block. If the previous block does not exist (as it is in the previous frame, so may not have been received), then the values at the start of the first block in the frame need to be defined. This is done by inserting an initialiser **audioBlockFormat** before the first block, with the ID set to ‘AB_xxxxxyyy_00000000’ and the **initializeBlock** attribute is set to ‘1’. This initialiser **audioBlockFormat** has no duration, so shall not contain an **Iduration** attribute.

The comparison between total time and local time, when converting from a non-serial **audioBlockFormat**, is shown in Figs 11 and 12. Both these show that recalculation of the **position** value is avoided by specifying time points outside of the frame. This allows the renderer (or any other processor of the metadata) to decide how to recalculate the positions.

FIGURE 11

Use case of audioBlockFormat using rtime and duration

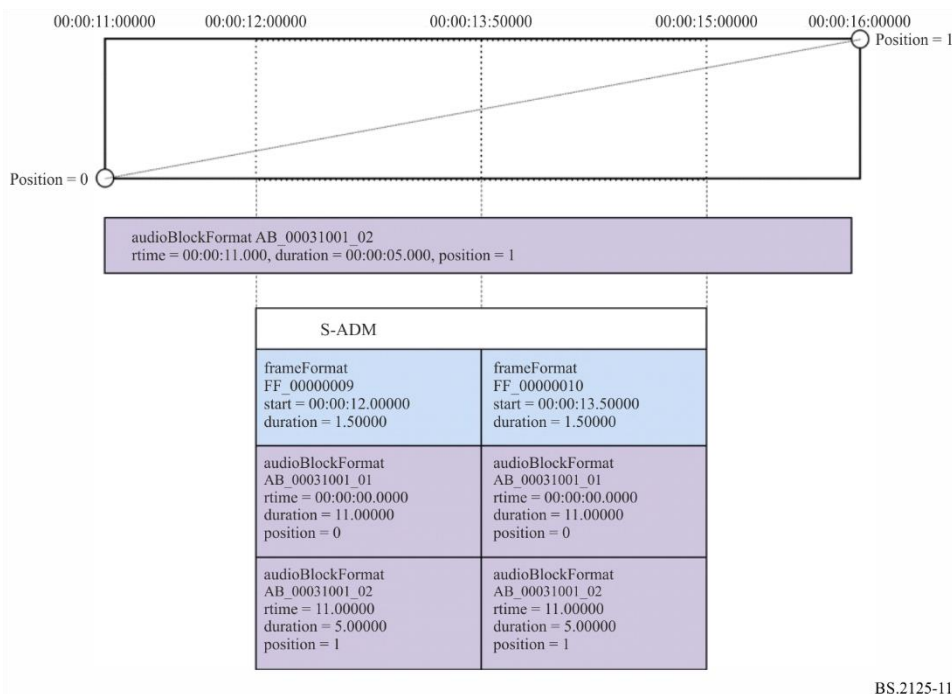
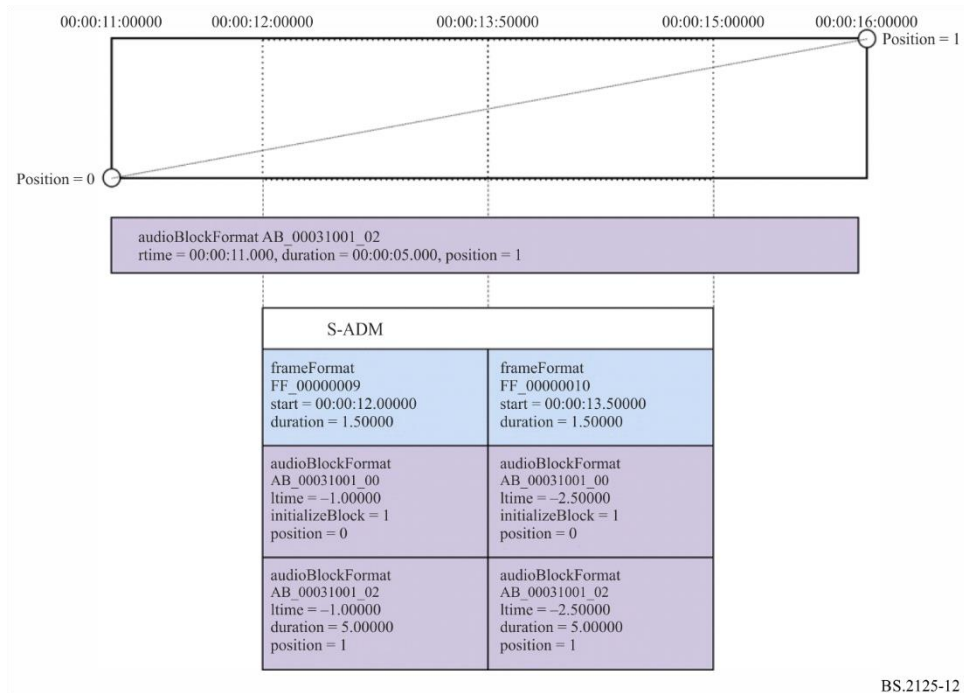


FIGURE 12

Use case of audioBlockFormat using ltime and lduration

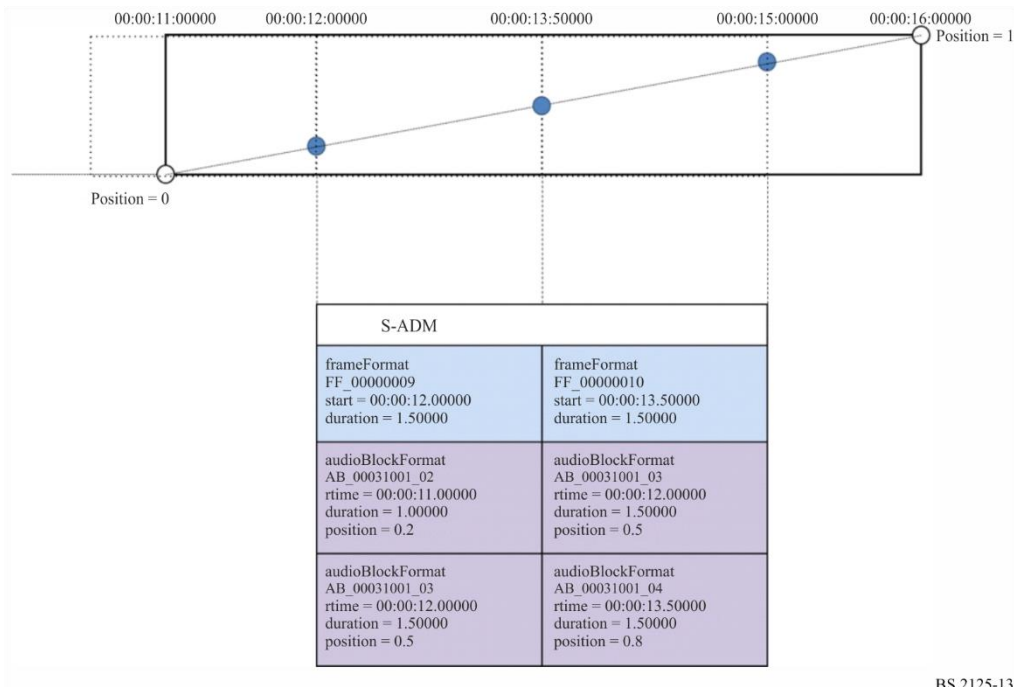


BS.2125-12

Figures 13 and 14 show how total time and local time are used when the S-ADM frames are generated from scratch. In this case the intermediate position values are known and already occur at the frame boundaries, so the **ltime** and **lduration** values occur within the frame.

FIGURE 13

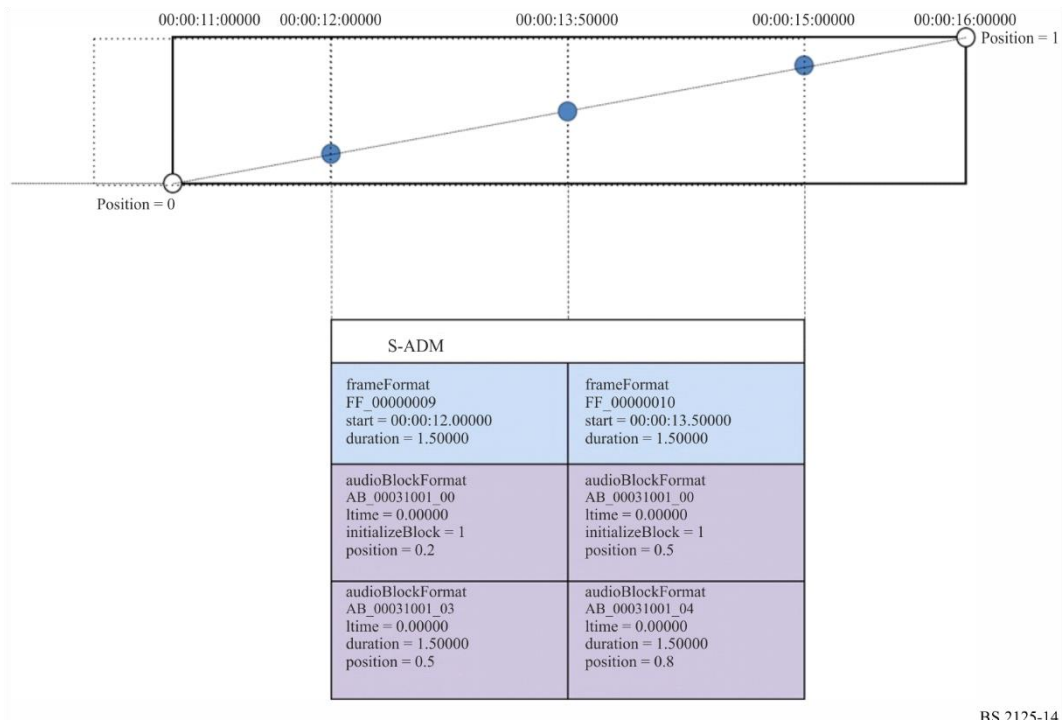
Use case of audioBlockFormat using rtime and duration when generated from scratch



BS.2125-13

FIGURE 14

Use case of audioBlockFormat using ltime and lduration when generated from scratch



BS.2125-14

A1.4.4.1 Additional attributes

TABLE 12

audioBlockFormat attributes

Attribute	Description	Example	Required
lstart	Start time of block relative to the start time of the S-ADM metadata frame. The start time of block is represented by one of these formats: <ul style="list-style-type: none"> – ‘hh:mm:ss.zzzzz’ which indicates hours, minutes, seconds and fractional seconds. – ‘ss.zzzzz’, if hours and minutes are not required – If ‘Sfffff’ is attached to the fractional part, the fractional part and the ‘fffff’ show the number of audio samples and the sampling rate, respectively. Note: the number of ‘z’ and ‘f’ digits can be from 5 to 9 digits, depending upon the desired precision.	– 00:00:00.00000 – 00.00000 – 00:00:00.00000S48000 – 0S48000	Optional

TABLE 12 (*end*)

Attribute	Description	Example	Required
Iduration	Duration of block in the S-ADM metadata frame. The duration of block is represented by one of these formats: <ul style="list-style-type: none"> – ‘hh:mm:ss.zzzzz’ which indicates hours, minutes, seconds and fractional seconds. – ‘ss.zzzzz’, if hours and minutes are not required – If ‘Sffff’ is attached to the fractional part, the fractional part and the ‘ffff’ show the number of audio samples and the sampling rate, respectively. Note: the number of ‘z’ and ‘f’ digits can be from 5 to 9 digits, depending upon the desired precision.	<ul style="list-style-type: none"> – 00:00:00.50000 – 00.50000 – 00:00:00.24000S48000 – 24000S48000 	Optional
initializeBlock	If the initializeBlock is set to ‘1’, it indicates the audioBlockFormat of ‘AB_xxxxxyyy_00000000’ is used to specify initial values of all elements for the first audio block in the frame.	1 (on), 0 (off)	Optional

A1.4.5 Compatibility with XML Broadcast Metadata

A1.4.5.1 Origin of Broadcast Metadata

The BWF file format (Recommendation ITU-R BS.1352) contains <bext> and <ubxt> chunks that carry broadcast metadata. This broadcast metadata can be carried in the BW64 file format (Recommendation ITU-R BS.2088) in the <axml> chunk alongside the ADM metadata. When the broadcast metadata is included in the XML in the <axml> chunk, the broadcast metadata parameters are located under the elements shown in Table 13.

TABLE 13

Element structure for including broadcast metadata

Level	Element	Broadcast metadata parameters
1 (top level)	coreMetadata	bextOriginator bextOriginatorReference bextDescription bextOriginationDate bextOriginationTime bextUMID
2	Format	bextCodingHistory
3	audioFormatExtended (contains ADM metadata)	bextTimeReference (within audioProgramme attributes)

A1.4.5.2 Broadcast Metadata in S-ADM

If broadcast metadata is to be included with the ADM metadata, then the element structure in Table 13 should be applied, with **coreMetadata** being the top-level element (below the **frame** element). The example XML in § A.1.6 illustrates this structure that includes broadcast metadata.

If broadcast metadata is being used in a S-ADM flow, it should only be used in ‘all’ or ‘header’ frames, whereas ‘full’, ‘divided’ and ‘intermediate’ frames retain the three elements in Table 13, but without the broadcast metadata parameters. This ensures each frame in the flow has the same element structure, but only the first frame contains the broadcast metadata (the second example XML in § A2.1 illustrates this structure without broadcast metadata).

When broadcast metadata is not being used in a S-ADM flow then either the top-level element (below the **frame** element) for the ADM metadata can be **audioFormatExtended**, or the three elements in Table 13 can be used (so **coreMetadata** containing **format**, then **audioFormatExtended**).

Annex 2

Example XML codes of S-ADM

A2.1 Example code for frame, frameHeader and audioFormatExtended

The following S-ADM XML code illustrates the structure of the top-level **frame** element, and the **frameHeader** and **audioFormatExtended** elements within it. Two frames are shown.

```
<frame>
  <frameHeader>
    <frameFormat frameFormatID = "FF_00000001" start = "00:00:00.00000" duration =
"00:00:00.50000" flowID = "12345678-abcd-4000-a000-112233445566" type = "header"/>
    <transportTrackFormat/>
  </frameHeader>

  <!-- the associated XML code of the ADM is written below -->
  <audioFormatExtended>
    <audioProgramme/>
    ...
    <audioChannelFormat/>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000002" start="00:00:00.50000" duration="00:00:00.50000"
flowID="12345678-abcd-4000-a000-112233445566" type="full"/>
    <transportTrackFormat/>
  </frameHeader>

  <!-- the associated XML code of the ADM is written below -->
```

```

<audioFormatExtended>
  <audioProgramme/>
  ...
  <audioChannelFormat/>
</audioFormatExtended>
</frame>

```

The following S-ADM XML code shows how the frames in the above code can be represented with **audioFormatExtended** inside parent elements. The parent elements are used in any frames of the S-ADM frames.

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID = "FF_00000001" start = "00:00:00.00000" duration =
"00:00:00.50000" flowID = "12345678-abcd-4000-a000-112233445566" type = "header"/>
    <transportTrackFormat/>
  </frameHeader>

  <!-- the associated XML code of the ADM is written below -->
  <coreMetadata>
    <format>
      <audioFormatExtended>
        <audioProgramme/>
        ...
        <audioChannelFormat/>
      </audioFormatExtended>
    </format>
  </coreMetadata>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000002" start="00:00:00.50000" duration="00:00:00.50000"
flowID="12345678-abcd-4000-a000-112233445566" type="full"/>
    <transportTrackFormat/>
  </frameHeader>

  <!-- the associated XML code of the ADM is written below -->
  <coreMetadata>
    <format>
      <audioFormatExtended>
        <audioProgramme/>
        ...
        <audioChannelFormat/>
      </audioFormatExtended>
    </format>
  </coreMetadata>
</frame>

```

A2.2 Example of usage of changedIDs

The example code below shows how three **audioChannelFormats** change over two frames ('FF_00000003' and 'FF_00000004'), and how the **status** attribute in the ID references in the **changedIDs** list is set:

- 'AC_00031001' first appears in 'FF_00000003' (therefore **status**='new'), and another **audioBlockFormat** is added to it in 'FF_00000004' (therefore **status**='changed');
- 'AC_00031002' is already established by 'FF_00000003' (therefore the ID reference is not listed), and disappears in 'FF_00000004' (therefore **status**='expired');
- 'AC_00031003' is already established by 'FF_00000003' (therefore the ID reference is not listed), and the **audioBlockFormat** in 'FF_00000004' has its duration increased (therefore **status**='extended').

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000003" start="00:00:02.00000" duration="00:00:01.00000"
type="full">
      <changedIDs>
        <audioChannelFormatIDRef status="new">AC_00031001</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
    ...
  </frameHeader>

  <audioFormatExtended>
    ...
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
    </audioChannelFormat>
    <audioChannelFormat audioChannelFormatID="AC_00031002">
      <audioBlockFormat audioBlockFormatID="AB_00031002_00000002" rtime="00:00:01.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">45.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
    </audioChannelFormat>
    <audioChannelFormat audioChannelFormatID="AC_00031003">
      <audioBlockFormat audioBlockFormatID="AB_00031003_00000002" rtime="00:00:01.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">90.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
    </audioChannelFormat>
    ...
  </audioFormatExtended>
</frame>

```

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000004" start="00:00:03.00000" duration="00:00:01.00000"
type= "full">
      <changedIDs>
        <audioChannelFormatIDRef status="changed">AC_00031001</audioChannelFormatIDRef>
        <audioChannelFormatIDRef status="expired">AC_00031002</audioChannelFormatIDRef>
        <audioChannelFormatIDRef status="extended">AC_00031003</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
    ...
  </frameHeader>

  <audioFormatExtended>
    ...
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:01.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">20.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
    </audioChannelFormat>
    <audioChannelFormat audioChannelFormatID="AC_00031003">
      <audioBlockFormat audioBlockFormatID="AB_00031003_00000002" rtime="00:00:01.00000"
duration="00:00:02.00000">
        <position coordinate= "azimuth">90.0</position>
        <position coordinate= "elevation">0.0</position>
      </audioBlockFormat>
    </audioChannelFormat>
    ...
  </audioFormatExtended>
</frame>

```

A2.3 Example of S-ADM XML derived from ADM XML.

This example shows how a set of S-ADM frames are derived from a single ADM XML file.

Sample of XML code for the original ADM is as follows:

```

<audioFormatExtended>
  <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
end="10:00:10.00000">
    <audioContentIDRef>ACO_1001</audioContentIDRef>
  </audioProgramme>
  <audioContent audioContentID="ACO_1001">
    <audioObjectIDRef>AO_1001</audioObjectIDRef>
  </audioContent>
</audioFormatExtended>

```

```

</audioContent>
<audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
</audioObject>
<audioPackFormat audioPackFormatID="AP_00031001">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioPackFormat>
<audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
</audioTrackUID>
<audioTrackFormat audioTrackFormatID="AT_00031001_01">
  <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
</audioTrackFormat>
<audioStreamFormat audioStreamFormatID="AS_00031001">
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioStreamFormat>
<audioChannelFormat audioChannelFormatID="AC_00031001">
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:03.00000">
    <position coordinate= "azimuth">30.0</position>
    <position coordinate= "elevation">0.0</position>
    <jumpPosition>1</jumpPosition>
  </audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
duration="00:00:03.00000">
    <position coordinate= "azimuth">-30.0</position>
    <position coordinate= "elevation">0.0</position>
    <jumpPosition>1</jumpPosition>
  </audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:06.00000"
duration="00:00:03.00000">
    <position coordinate= "azimuth">0.0</position>
    <position coordinate= "elevation">0.0</position>
    <jumpPosition>0</jumpPosition>
  </audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000004" rtime="00:00:09.00000"
duration="00:00:01.00000">
    <position coordinate= "azimuth">30.0</position>
    <position coordinate= "elevation">0.0</position>
    <jumpPosition>0</jumpPosition>
  </audioBlockFormat>
</audioChannelFormat>
</audioFormatExtended>

```

Samples of XML code for the S-ADM with a 1.5 second frame size, and MF flow, are as follows:

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000001" start="10:00:00.00000" duration="00:00:01.50000" type=
"header"/>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
end="10:00:10.00000">
      <audioContentIDRef>ACO_1001</audioContentIDRef>
    </audioProgramme>
    <audioContent audioContentID="ACO_1001">
      <audioObjectIDRef>AO_1001</audioObjectIDRef>
    </audioContent>
    <audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
    </audioObject>
    <audioPackFormat audioPackFormatID="AP_00031001">
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioPackFormat>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:03.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
    <audioStreamFormat audioStreamFormatID="AS_00031001">
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioStreamFormat>
    <audioTrackFormat audioTrackFormatID="AT_00031001_01">
      <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
    </audioTrackFormat>
    <audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
    </audioTrackUID>
  </audioFormatExtended>
</frame>
<frame>

```

```

<frameHeader>
  <frameFormat frameFormatID="FF_00000002" start="10:00:01.50000" duration="00:00:01.50000"
type="intermediate" countToFull="3"/>
</frameHeader>

<audioFormatExtended/>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000003" start="10:00:03.00000" duration="00:00:01.50000"
type="intermediate" countToFull="2">
      <changedIDs>
        <audioChannelFormatIDRef status="changed">AC_00031001</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
duration="00:00:03.00000">
        <position coordinate="azimuth">-30.0</position>
        <position coordinate="elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000004" start="10:00:04.50000" duration="00:00:01.50000"
type="intermediate" countToFull="1"/>
  </frameHeader>

  <audioFormatExtended/>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000005" start="10:00:06.00000" duration="00:00:01.50000"
type="full">
      <changedIDs>
        <audioChannelFormatIDRef status="changed"> AC_00031001</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>

```



```

</frameHeader>

<audioFormatExtended>
  <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
end="10:00:10.00000">
  <audioContentIDRef>ACO_1001</audioContentIDRef>
</audioProgramme>
<audioContent audioContentID="ACO_1001">
  <audioObjectIDRef>AO_1001</audioObjectIDRef>
</audioContent>
<audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
</audioObject>
<audioPackFormat audioPackFormatID="AP_00031001">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioPackFormat>
<audioChannelFormat audioChannelFormatID="AC_00031001">
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
duration="00:00:03.00000">
  <position coordinate= "azimuth">-30.0</position>
  <position coordinate= "elevation">0.0</position>
  <jumpPosition>1</jumpPosition>
</audioBlockFormat>
  <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:06.00000"
duration="00:00:03.00000">
  <position coordinate= "azimuth">0.0</position>
  <position coordinate= "elevation">0.0</position>
  <jumpPosition>0</jumpPosition>
</audioBlockFormat>
</audioChannelFormat>
<audioStreamFormat audioStreamFormatID="AS_00031001">
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioStreamFormat>
<audioTrackFormat audioTrackFormatID="AT_00031001_01">
  <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
</audioTrackFormat>
<audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
</audioTrackUID>
</audioFormatExtended>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000006" start="10:00:07.50000" duration="00:00:01.50000"
type="intermediate" countToFull="3"/>
  </frameHeader>

```

```

<audioFormatExtended/>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000007" start="10:00:09.00000" duration="00:00:01.00000"
type="intermediate" countToFull="2">
      <changedIDs>
        <audioChannelFormatIDRef status="changed">AC_00031001</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000004" rtime="00:00:09.00000"
duration="00:00:01.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>0</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

```

Samples of XML code for the S-ADM with a 1.5 second frame size, and DF flow, are as follows:

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000001_01" start="10:00:00.00000"
duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioProgramme</chunkAdmElement>
      <chunkAdmElement>audioContent</chunkAdmElement>
      <chunkAdmElement>audioObject</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
end="10:00:10.00000">
      <audioContentIDRef>ACO_1001</audioContentIDRef>
    </audioProgramme>
    <audioContent audioContentID="ACO_1001">

```

```

    <audioObjectIDRef>AO_1001</audioObjectIDRef>
  </audioContent>
  <audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
    <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
    <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
  </audioObject>
</audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000001_02" start="10:00:00.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="2">
      <chunkAdmElement>audioPackFormat</chunkAdmElement>
      <chunkAdmElement>audioStreamFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioPackFormat audioPackFormatID="AP_00031001">
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioPackFormat>
    <audioStreamFormat audioStreamFormatID="AS_00031001">
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioStreamFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000001_03" start="10:00:00.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioTrackFormat</chunkAdmElement>
      <chunkAdmElement>audioTrackUID</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioTrackFormat audioTrackFormatID="AT_00031001_01">
      <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
    </audioTrackFormat>
    <audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
    </audioTrackUID>
  </audioFormatExtended>

```

```

</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_0000001_04" start="10:00:00.00000"
      duration="00:00:01.50000" type= "divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
        duration="00:00:03.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_0000002_01" start="10:00:01.50000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioProgramme</chunkAdmElement>
      <chunkAdmElement>audioContent</chunkAdmElement>
      <chunkAdmElement>audioObject</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
      end="10:00:10.00000">
      <audioContentIDRef>ACO_1001</audioContentIDRef>
    </audioProgramme>
    <audioContent audioContentID="ACO_1001">
      <audioObjectIDRef>AO_1001</audioObjectIDRef>
    </audioContent>
    <audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
    </audioObject>
  </audioFormatExtended>

```

```

    </audioObject>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000002_04" start="10:00:01.50000"
      duration="00:00:01.50000" type= "divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
        duration="00:00:03.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000003_02" start="10:00:03.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioPackFormat</chunkAdmElement>
      <chunkAdmElement>audioStreamFormat</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioPackFormat audioPackFormatID="AP_00031001">
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioPackFormat>
    <audioStreamFormat audioStreamFormatID="AS_00031001">
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioStreamFormat>
  </audioFormatExtended>
</frame>

```

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000003_04" start="10:00:03.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
        duration="00:00:03.00000">
        <position coordinate="azimuth">-30.0</position>
        <position coordinate="elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000004_03" start="10:00:04.50000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioTrackFormat</chunkAdmElement>
      <chunkAdmElement>audioTrackUID</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioTrackFormat audioTrackFormatID="AT_00031001_01">
      <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
    </audioTrackFormat>
    <audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
    </audioTrackUID>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>

```

```

<frameFormat frameFormatID="FF_00000004_04" start="10:00:04.50000"
  duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="1">
  <chunkAdmElement>audioChannelFormat</chunkAdmElement>
  <chunkAdmElement>audioBlockFormat</chunkAdmElement>
</frameFormat>
</frameHeader>

<audioFormatExtended>
  <audioChannelFormat audioChannelFormatID="AC_00031001">
    <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
      duration="00:00:03.00000">
      <position coordinate="azimuth">-30.0</position>
      <position coordinate="elevation">0.0</position>
      <jumpPosition>1</jumpPosition>
    </audioBlockFormat>
  </audioChannelFormat>
</audioFormatExtended>
</frame>
<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000005_01" start="10:00:06.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioProgramme</chunkAdmElement>
      <chunkAdmElement>audioContent</chunkAdmElement>
      <chunkAdmElement>audioObject</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioProgramme audioProgrammeID="APR_1001" audioProgrammeName="Main" start="10:00:00.00000"
      end="10:00:10.00000">
      <audioContentIDRef>ACO_1001</audioContentIDRef>
    </audioProgramme>
    <audioContent audioContentID="ACO_1001">
      <audioObjectIDRef>AO_1001</audioObjectIDRef>
    </audioContent>
    <audioObject audioObjectID="AO_1001" start="00:00:00.00000" duration="00:00:10.00000">
      <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
      <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
    </audioObject>
  </audioFormatExtended>
</frame>

```



```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000005_04" start="10:00:06.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
        duration="00:00:03.00000">
        <position coordinate="azimuth">-30.0</position>
        <position coordinate="elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:06.00000"
        duration="00:00:03.00000">
        <position coordinate="azimuth">0.0</position>
        <position coordinate="elevation">0.0</position>
        <jumpPosition>0</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat
      frameFormatID="FF_00000006_02" start="10:00:07.50000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioPackFormat</chunkAdmElement>
      <chunkAdmElement>audioStreamFormat</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioPackFormat audioPackFormatID="AP_00031001">
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioPackFormat>
    <audioStreamFormat audioStreamFormatID="AS_00031001">
      <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
      <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
    </audioStreamFormat>
  </audioFormatExtended>

```

```

    </audioStreamFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000006_04" start="10:00:07.50000"
      duration="00:00:01.50000" type= "divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" rtime="00:00:03.00000"
        duration="00:00:03.00000">
        <position coordinate= "azimuth">-30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>1</jumpPosition>
      </audioBlockFormat>
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:06.00000"
        duration="00:00:03.00000">
        <position coordinate= "azimuth">0.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>0</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000007_03" start="10:00:09.00000"
      duration="00:00:01.50000" type="divided" numMetadataChunks="4" countToSameChunk="3">
      <chunkAdmElement>audioTrackFormat</chunkAdmElement>
      <chunkAdmElement>audioTrackUID</chunkAdmElement>
    </frameFormat>
    <transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="1" numTracks="1">
      <audioTrack trackID="1">
        <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
      </audioTrack>
    </transportTrackFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioTrackFormat audioTrackFormatID="AT_00031001_01">
      <audioStreamFormatIDRef>AS_00031001</audioStreamFormatIDRef>
    </audioTrackFormat>

```

```

<audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
</audioTrackUID>
</audioFormatExtended>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000007_04" start="10:00:09.00000"
      duration="00:00:01.50000" type= "divided" numMetadataChunks="4" countToSameChunk="1">
      <chunkAdmElement>audioChannelFormat</chunkAdmElement>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
    <audioChannelFormat audioChannelFormatID="AC_00031001">
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000003" rtime="00:00:06.00000"
        duration="00:00:03.00000">
        <position coordinate= "azimuth">0.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>0</jumpPosition>
      </audioBlockFormat>
      <audioBlockFormat audioBlockFormatID="AB_00031001_00000004" rtime="00:00:09.00000"
        duration="00:00:01.00000">
        <position coordinate= "azimuth">30.0</position>
        <position coordinate= "elevation">0.0</position>
        <jumpPosition>0</jumpPosition>
      </audioBlockFormat>
    </audioChannelFormat>
  </audioFormatExtended>
</frame>

```

A2.4 Example of transportTrackFormat use

This example shows how the **transportTrackFormat** is equivalent to the ‘chna’ chunk in a BW64 file.

Example of the metadata in the ‘chna’ chunk of the BW64 is as follows:

```

ckID = {'c', 'h', 'n', 'a'};
ckSize = 164;
numTracks = 3;
numUIDs = 4;
ID[0] = {trackIndex=1; UID="ATU_00000001"; trackRef="AT_00031001_01"; packRef="AP_00031001"};
ID[1] = {trackIndex=1; UID="ATU_00000002"; trackRef="AT_00031002_01"; packRef="AP_00031002"};
ID[2] = {trackIndex=2; UID="ATU_00000003"; trackRef="AT_00031003_01"; packRef="AP_00031001"};
ID[3] = {trackIndex=3; UID="ATU_00000004"; trackRef="AT_00031004_01"; packRef="AP_00031003"};

```

In case of conveying the above audio signals over two AES-3 interfaces, example of the **transportTrackFormat** and the **audioTrackUID** is as follows:

```

<!-- ##### -->
<!-- Transport Track -->
<!-- ##### -->
<transportTrackFormat transportID="TP_0001" transportName="AES3-A" numIDs="3" numTracks="2">
  <audioTrack trackID="1">
    <audioTrackUIDRef>ATU_00000001</audioTrackUIDRef>
    <audioTrackUIDRef>ATU_00000002</audioTrackUIDRef>
  </audioTrack>
  <audioTrack trackID="2">
    <audioTrackUIDRef>ATU_00000003</audioTrackUIDRef>
  </audioTrack>
</transportTrackFormat>
<transportTrackFormat transportID="TP_0002" transportName="AES3-B" numIDs="1" numTracks="1">
  <audioTrack trackID="1">
    <audioTrackUIDRef>ATU_00000004</audioTrackUIDRef>
  </audioTrack>
</transportTrackFormat>

<!-- ##### -->
<!-- Audio Track UIDs -->
<!-- ##### -->
<audioFormatExtended>
  ...
  <audioTrackUID UID="ATU_00000001" sampleRate="48000" bitDepth="24">
    <audioTrackFormatIDRef>AT_00031001_01</audioTrackFormatIDRef>
    <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  </audioTrackUID>
  <audioTrackUID UID="ATU_00000002" sampleRate="48000" bitDepth="24">
    <audioTrackFormatIDRef>AT_00031002_01</audioTrackFormatIDRef>
    <audioPackFormatIDRef>AP_00031002</audioPackFormatIDRef>
  </audioTrackUID>
  <audioTrackUID UID="ATU_00000003" sampleRate="48000" bitDepth="24">
    <audioTrackFormatIDRef>AT_00031003_01</audioTrackFormatIDRef>
    <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  </audioTrackUID>
  <audioTrackUID UID="ATU_00000004" sampleRate="48000" bitDepth="24">
    <audioTrackFormatIDRef>AT_00031004_01</audioTrackFormatIDRef>
    <audioPackFormatIDRef>AP_00031003</audioPackFormatIDRef>
  </audioTrackUID>
  ...
</audioFormatExtended>

```

A2.5 Example of how the timeReference flag is used

This example shows how ADM XML can be converted to S-ADM in either ‘total’ or ‘local’ **timeReference** mode. The **audioBlockFormat** timing parameters (**rtime** and **duration**) from the original ADM are retained in the ‘total’ version of the S-ADM XML; in the ‘local’ version, these timing parameters are converted to **lstart** and **lduration**.

Example of the XML code described in the original ADM is as follows (some attributes and elements are not included to improve clarity):

```
<audioFormatExtended>
...
<audioObject audioObjectID="AO_1001" start="00:00:01.00000">
  <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
</audioObject>
<audioPackFormat audioPackFormatID="AP_00031001">
  <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
</audioPackFormat>
<audioChannelFormat audioChannelFormatID="AC_00031001">
  <audioBlockFormat      audioBlockFormatID="AB_00031001_00000001"      rtime="00:00:00.00000"
duration="00:00:01.00000">
    <position coordinate= "azimuth">30.0</position>
    <position coordinate= "elevation">0.0</position>
    <jumpPosition>1</jumpPosition>
  </audioBlockFormat>
  <audioBlockFormat      audioBlockFormatID="AB_00031001_00000002"      rtime="00:00:01.00000"
duration="00:00:01.00000">
    <position coordinate="azimuth">0.0</position>
    <position coordinate="elevation">0.0</position>
    <jumpPosition>0</jumpPosition>
  </audioBlockFormat>
</audioChannelFormat>
...
</audioFormatExtended>
```

The following example is written in the S-ADM with ‘total’ time. The **rtime** in the **audioBlockFormat** is relative to the start time of the referencing **audioObject**:

```
<frame>
  <frameHeader>
    <frameFormat  frameFormatID="FF_00000003"  start="00:00:01.00000"  duration="00:00:00.50000"
timeReference= "total" type="full">
    <changedIDs>
      <audioChannelFormatIDRef status="changed">AC_00031001</audioChannelFormatIDRef>
    </changedIDs>
  </frameFormat>
</frameHeader>

<audioFormatExtended>
...
<audioObject audioObjectID="AO_1001" start="00:00:01.00000">
```

```

    <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  </audioObject>
  <audioPackFormat audioPackFormatID="AP_00031001">
    <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
  </audioPackFormat>
  <audioChannelFormat audioChannelFormatID="AC_00031001">
    <audioBlockFormat audioBlockFormatID="AB_00031001_00000001" rtime="00:00:00.00000"
duration="00:00:01.00000">
      <position coordinate="azimuth">30.0</position>
      <position coordinate="elevation">0.0</position>
      <jumpPosition>1</jumpPosition>
    </audioBlockFormat>
  </audioChannelFormat>
  ...
</audioFormatExtended>
</frame>

```

The following example is written in the S-ADM with ‘local’ time. As **ltime** in the **audioBlockFormat** is relative to the frame start time, the referencing **audioObject** does not require a start time:

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000003" start="00:00:01.00000" duration="00:00:00.50000"
timeReference="local" type="full">
      <changedIDs>
        <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
      </changedIDs>
    </frameFormat>
  </frameHeader>

  <audioFormatExtended>
  ...
  <audioObject audioObjectID="AO_1001">
    <audioPackFormatIDRef>AP_00031001</audioPackFormatIDRef>
  </audioObject>
  <audioPackFormat audioPackFormatID="AP_00031001">
    <audioChannelFormatIDRef>AC_00031001</audioChannelFormatIDRef>
  </audioPackFormat>
  <audioChannelFormat audioChannelFormatID="AC_00031001">
    <audioBlockFormat audioBlockFormatID="AB_00031001_00000000" initializeBlock="1">
      <position coordinate="azimuth">30.0</position>
      <position coordinate="elevation">0.0</position>
    </audioBlockFormat>
    <audioBlockFormat audioBlockFormatID="AB_00031001_00000002" ltime="00:00:00.00000"
lduration="00:00:00.50000">
      <position coordinate="azimuth">15.0</position>
      <position coordinate="elevation">0.0</position>
      <jumpPosition>0</jumpPosition>
    </audioBlockFormat>
  </audioChannelFormat>

```

```

    </audioChannelFormat>
    ...
  </audioFormatExtended>
</frame>

```

A2.6 Example of how the broadcast metadata is used

This example shows how the broadcast metadata is used.

```

<frame>
  <frameHeader>
    <frameFormat frameFormatID = "FF_00000001" start = "00:00:00.00000" duration =
"00:00:00.50000" flowID = "12345678-abcd-4000-a000-112233445566" type = "header"/>
    <transportTrackFormat/>
  </frameHeader>

  <coreMetadata>
    <date>
      <created statDate="2000-10-10" startTime="12:00:00"/>
    </date>
    <format>
      <audioFormatExtended>
        <!--the XML code of the ADM is written here -->
        <audioProgramme/>
        ...
        <audioChannelFormat/>
      </audioFormatExtended>
    </format>
  </coreMetadata>
</frame>

<frame>
  <frameHeader>
    <frameFormat frameFormatID="FF_00000002" start="00:00:00.50000" duration="00:00:00.50000"
flowID="12345678-abcd-4000-a000-112233445566" type="full"/>
    <transportTrackFormat/>
  </frameHeader>

  <coreMetadata>
    <format>
      <audioFormatExtended>
        <!--the XML code of the ADM is written here -->
        <audioProgramme/>
        ...
        <audioChannelFormat/>
      </audioFormatExtended>
    </format>
  </coreMetadata>
</frame>

```
