SMPTE REGISTERED DISCLOSURE DOCUMENT

Dolby Atmos® Print Master File Specification

The attached document is a Registered Disclosure Document prepared by the proponent identified below. It has been examined by the appropriate SMPTE Technology Committee and is believed to contain adequate information to satisfy the objectives defined in the Scope, and to be technically consistent.

This document is NOT a Standard, Recommended Practice or Engineering Guideline, and does NOT imply a finding or representation of the Society.

Errors in this document should be reported to the proponent identified below, with a copy to eng@smpte.org.

This document is intended to allow the interpretation of Dolby Atmos Print Master files. It is not intended to support the development of hardware or software applications that create or process these files. Creation and processing of such files is reserved to individuals and organizations that have entered into agreements with the proponent identified below for this purpose. Use of this document to produce or process Dolby Atmos files using non-Dolby tools would potentially cause user confusion, diminished sound quality as experienced by content consumers, and damage to the reputation of the Dolby Atmos brand and to Dolby Laboratories itself.

All other inquiries in respect of this document, including inquiries as to intellectual property requirements that may be attached to use of the disclosed technology, should be addressed to the proponent identified below.

Proponent contact information:

Dean Bullock
Dolby Laboratories Inc.
100 Potrero Ave.
San Francisco, CA 94103

Email dcb@dolby.com

Approved
June 17, 2014

Table of Contents

Introduction

Dolby Atmos® is an advanced cinema sound format comprising an audio essence and metadata stream played through specialized renderers in the cinema.

1 Scope

This document defines how to store the audio essence and metadata for a Dolby Atmos presentation. The files created with this specification are part of the DCDM and are passed on to other processes to package for distribution and playback.

2 Normative References

The following standards contain provisions that, through reference in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this recommended practice are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

[XML Namespaces] World Wide Web Consortium (W3C) (2009, December 8). Namespaces in XML 1.0 (Third Edition)

[XML 1.1 Schema Part 1: Structures] World Wide Web Consortium (W3C) (2012, April 5). XML Schema Definition Language (XSD) 1.1 Part 1: Structures.

[XML 1.1 Schema Part 2: Datatypes] World Wide Web Consortium (W3C) (2012, April 5). XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes.

[XML] World Wide Web Consortium (W3C) (2008, November 26) Extensible Markup Language (XML) 1.0 (Fifth Edition)

[BWAV] ITU-R BR.1352-2:2002, Broadcast Wave Format

3 Overview

Dolby Atmos is a hybrid audio format that utilizes both channel-based and object-based audio. A Dolby Atmos print master needs to contain information to support both of these, and therefore has several components:

- Up to ten mono PCM audio files for any of the ten bed channels (traditional 7.1, plus a stereo overhead).
- Up to 118 mono PCM audio files, one for each object.
- One .PRM metadata file for each object. Each audio object also requires a metadata file that describes how it should play back. This is a space-delimited text file that can be opened and edited in a text editor.
- One .RPL file. This file contains program-level information about the print master, including object count, file paths, and start time offset. This is an .XML file that can be opened and edited in a text or XML editor.

4 Audio Essence File

The audio files store the audio essence associated with a bed channel or an object. Each file shall be a mono Broadcast Wave file [BWAV]. The filename of the bed channel file should carry the channel id in the filename (see Section 5.3.1). The filename of the audio object file should carry the objectStream id in the filename (see Section 5.4).

Each audio file in a print master shall be continuous and have the same duration, which is the duration of the print master. There is not a defined relationship between the duration of the print master and the duration of a composition or reel.

The sample rate of the audio file shall be 48 kHz.

5 Render Playlist File

The Render Playlist (RPL) file describes what is included in the Dolby Atmos print master. Dolby Atmos is a flexible format that does not require a specific number of audio objects, so each print master folder can contain a different number of essence and metadata files. The RPL file describes and links these files together to create the print master.

The RPL file is an XML file [XML]. The elements in the file are described below and the schema for the file is specified in Section 7.

5.1 playlist

This element is the top level of the RPL file. It contains the other elements that make up the playlist.

5.2 offset

The offset value represents the duration in seconds between 00:00:00:00 and the start of the print master file. For example, at 24 fps, if the first sample in the file is associated with the time 00:59:57:23, then the RPL offset would be: (59 * 60) + 57 + (23/24) = 3597.95833 seconds. The value is stored as a decimal number. The offset value is derived from the time code present when the first sample is stored and it is assumed that the picture has been synchronized to the same time code.

5.3 bed

The optional bed element carries the information about the bed channels. It is an unordered list of ids and channels. The sub-elements are described below.

5.3.1 channel

The channel element carries information about the bed channel that supports identification of its intended reproduction location and the location of the essence file. It has two optional elements: id and audioPath. The number of channel elements in a playlist must be 10 or less.

5.3.2 id

The id element is a string that carries the intended reproduction location. The string value and the associated speaker location are shown in the table below.

Id	Associated speaker location
L	Left Screen Speaker
R	Right Screen Speaker
C	Center Screen Speaker
LFE	Low Frequency Effects Speaker
Lss	Left Side Surround Array
Rss	Right Side Surround Array
Lb	Left Back Surround Array
Rb	Right Back Surround Array
Lc	Left Ceiling Array
Rc	Right Ceiling Array

5.3.3 audioPath

The audioPath element is a string that carries the path to the essence file associated with this bed channel. The format of the string is defined in Section 5.4.4 .

5.4 objectStream

The objectStream element carries information about the audio object. The number of objectStream elements shall be 118 or less.

5.4.1 id

The id is a text string that allows identifying the audio object. A playlist shall not carry two or more ObjectStreams with matching ids. A playlist shall not have an id value greater than 117.

5.4.2 audioPath

The audioPath element is a string that carries the path to the essence file associated with this audio object. The format of the string is defined in Section 5.4.4.

5.4.3 metadataPath

The metadataPath element is a string that carries the path to the metadata (PRM) file associated with this audio object. The format of the string is defined in Section 5.4.4.

5.4.4 File Paths

All file path elements such as the audioPath and metadataPath elements shall consist of 8-bit ASCII characters from the set a-z, A-Z, 0-9, “-“ (dash), “_”(underscore) and “.” (period). The path shall be optionally divided into segments using the “/” (slash) character. No segment shall have more than 128 characters, and no path value shall exceed 1024 characters in length. A path value shall have no more than 10 segments. The path value shall preserve case (the path and the filename on the file system shall have identical case). The path value shall not begin with a ‘/’ (slash) character; i.e. all paths shall be relative to the directory which contains the RPL file.

Example RPL File

Figure 1 provides an example of an RPL file. In this example, there are the standard ten bed channels, and six objects (with IDs 1, 2, 3, 4, 33, and 34).

```
<?xml version="1.0" encoding="utf-8"?>
<playlist>
<offset>3597.95833</offset>
<bed>
   <channel><audioPath>dub_out_Bed.L.wav</audioPath><id>L</id></channel>
   <channel><audioPath>dub_out_Bed.R.wav</audioPath><id>R</id></channel>
   <channel><audioPath>dub_out_Bed.C.wav</audioPath><id>C</id></channel>
   <channel><audioPath>dub_out_Bed.LFE.wav</audioPath><id>LFE</id></channel>
   <channel><audioPath>dub_out_Bed.Lss.wav</audioPath><id>Lss</id></channel>
   <channel><audioPath>dub_out_Bed.Rss.wav</audioPath><id>Rss</id></channel>
   <channel><audioPath>dub_out_Bed.Lb.wav</audioPath><id>Lb</id></channel>
   <channel><audioPath>dub_out_Bed.Rb.wav</audioPath><id>Rb</id></channel>
   <channel><audioPath>dub_out_Bed.Lc.wav</audioPath><id>Lc</id></channel>
   <channel><audioPath>dub_out_Bed.Rc.wav</audioPath><id>Rc</id></channel>
</bed>
<objectStream>
   <id>1</id>
   <audioPath>dub_out_Obj_1.wav</audioPath>
   <metadataPath>dub_out_Obj_1.prm</metadataPath>
</objectStream>
<objectStream>
   <id>2</id>
   <audioPath>dub_out_Obj_2.wav</audioPath>
   <metadataPath>dub_out_Obj_2.prm</metadataPath>
</objectStream>
<objectStream>
   <id>3</id>
   <audioPath>dub_out_Obj_3.wav</audioPath>
   <metadataPath>dub_out_Obj_3.prm</metadataPath>
</objectStream>
<objectStream>
   <id>4</id>
   <audioPath>dub_out_Obj_4.wav</audioPath>
   <metadataPath>dub_out_Obj_4.prm</metadataPath>
</objectStream>
<objectStream>
   <id>33</id>
   <audioPath>dub_out_Obj_33.wav</audioPath>
   <metadataPath>dub_out_Obj_33.prm</metadataPath>
</objectStream>
<objectStream>
   <id>34</id>
   <audioPath>dub_out_Obj_34.wav</audioPath>
   <metadataPath>dub_out_Obj_34.prm</metadataPath>
</objectStream>
</playlist>
```

Figure 1 – Example RPL File

6 PRM File

The .PRM file is associated with a specific audio object (as defined by the RPL file) and contains the metadata needed to render the audio object. The PRM file is a text file with fields of coded text. The fields are specified below.

6.1 Fields

The following fields represent various positional and decoding parameters associated with each object. Each line shall contain all of the following fields in the order specified below. Each field shall be separated by a single space. The X, Y, and Z positional values use a room-based reference with the front of the room defined by the placement of the viewing screen. The bottom of the room is defined by the lowest speaker placement:

- Timestamp (in seconds): The number of seconds from the start of the print master file that a metadata event occurs.
- X position: The left/right location of the object, represented by a floating-point value between 0 and 1, where 0 = left wall, and 1 = right wall. X positions within the room are linearly interpolated.
- Y position: The front/back location of the object, represented by a floating-point value between 0 and 1, where 0 = front wall (screen), and 1 = back wall. Y positions within the room are linearly interpolated.
- Z position: The top/bottom location of the object, represented by a floating-point value between 0 and 1, where 0 = bottom, and 1 = top. Z positions within the room are linearly interpolated.
- Object Render Mode: This field combines codes associated with several parameters for efficiency. It is represented as a single integer. See Section 6.1.1.
- Size: The spread of audio for an object across the room, represented by a floating-point value between 0 and 1, with zero being a point source and 1 being an object that fills the room.
- Reserved: This field is always 0.
- Decorrelation enable flag: When enabled, each speaker signal generated from the source object signal will pass through a decorrelation filter. 0 = disabled, and 1 = enabled.

6.1.1 Object Render Mode

The Object Render Mode field is calculated from the following parameters:

6.1.1.1 Zone Select

The Zone select parameter controls which zones an object will be rendered to. It may have one of the following values:

Zone select value	Enabled zones
0	All zones enabled
1	All zones except rear zones
2	All zones except side zones
3	Screen center and rear zones only
4	Screen only
5	Side and rear zones only – no screen

6.1.1.2 Speaker Snap Mode

If speaker snap is enabled, an object will be rendered at the speaker position that is closest to the location indicated by the positional metadata.

6.1.1.3 Elevation Enable Mode

An object will be rendered to overhead speakers only if elevation is enabled. Otherwise, it will be rendered to the base/side speakers only.

6.1.1.4 Object Render Mode Calculation

The object render mode field value is computed as follows:

60 + ([Zone Select] * 2) + (20, if Speaker Snap Mode is enabled) + (1, if Elevation Mode is enabled)

6.2 Example Line

0.005333 0.500000 0.329787 0.451000 81 0.000000 0 1

The preceding example represents the following data:

- Time stamp = 0.005333 seconds from print master start
- X position = 0.5
- Y position = 0.329787
- Z position = 0.451000
- Object Render Mode = 81, computed from 60 + (0 * 2) + 20 + 1:
  - Zone select = 0 (all zones)
  - Speaker snap mode = on
  - Elevation enable mode = on
- Size = 0
- Decorrelation = on

7 RPL File Schema

The RPL file shall be associated with a unique XML namespace name [XML Namespaces]. The namespace name shall be the string value "`http://dolby.com/AtmosMasterRpl`" This namespace name conveys both structural and semantic version information, and serves the purpose of a traditional version number field.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:vc="http://www.w3.org/2007/XMLSchema-versioning"
        xmlns="http://dolby.com/AtmosMasterRpl"
        targetNamespace="http://dolby.com/AtmosMasterRpl"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified"
        vc:minVersion="1.1">
   <xs:simpleType name="UUID">
     <xs:restriction base="xs:anyURI">
       <xs:pattern value="urn:uuid:[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-
[0-9a-fA-F]{12}"/>
     </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="offsetType">
      <xs:restriction base="xs:decimal">
         <xs:minInclusive value="0"/>
```

```
   </xs:restriction>
</xs:simpleType>
<xs:simpleType name="objectStreamIdType">
   <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="117"/>
   </xs:restriction>
</xs:simpleType>
<xs:simpleType name="bedChannelIdType">
   <xs:annotation>
      <xs:documentation>Bed channel identifier</xs:documentation>
   </xs:annotation>
   <xs:restriction base="xs:string">
      <xs:enumeration value="L">
         <xs:annotation>
            <xs:documentation>Left</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="R">
         <xs:annotation>
            <xs:documentation>Right</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="C">
         <xs:annotation>
            <xs:documentation>Center</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="LFE">
         <xs:annotation>
            <xs:documentation>Low Frequency Effect</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Lss">
         <xs:annotation>
            <xs:documentation>Left Side Surround</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Rss">
         <xs:annotation>
            <xs:documentation>Right Side Surround</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Lb">
         <xs:annotation>
            <xs:documentation>Left Back</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Rb">
         <xs:annotation>
            <xs:documentation>Right Back</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Lc">
         <xs:annotation>
            <xs:documentation>Left Ceiling</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="Rc">
         <xs:annotation>
            <xs:documentation>Right Ceiling</xs:documentation>
         </xs:annotation>
      </xs:enumeration>
   </xs:restriction>
</xs:simpleType>
<xs:element name="playlist">
   <xs:complexType>
      <xs:all>
         <xs:element name="playlistId" type="UUID" minOccurs=”0”/>
         <xs:element name="offset" type="offsetType"/>
         <xs:element name="bed" minOccurs="0">
```

```
               <xs:complexType>
                  <xs:sequence minOccurs="0" maxOccurs="unbounded">
                     <xs:element name="channel" maxOccurs="10">
                        <xs:complexType>
                           <xs:all>
                              <xs:element name="id"
                                    type="bedChannelIdType" minOccurs="1" maxOccurs="1"/>
                              <xs:element name="audioPath"
                                    type="xs:string" minOccurs="1" maxOccurs="1"/>
                           </xs:all>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name="objectStream" minOccurs="0" maxOccurs="118">
               <xs:complexType>
                  <xs:all>
                     <xs:element name="id" type="objectStreamIdType" minOccurs="0"/>
                     <xs:element name="audioPath" type="xs:string" minOccurs="1" maxOccurs="1"/>
                     <xs:element name="metadataPath" type="xs:string" minOccurs="1" maxOccurs="1"/>
                  </xs:all>
               </xs:complexType>
            </xs:element>
         </xs:all>
      </xs:complexType>
   </xs:element>
</xs:schema>
```