SMPTE REGISTERED DISCLOSURE DOCUMENT

MDA Bitstream 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.

Every attempt has been made to ensure that the information contained in this document is accurate. Errors in this document should be reported to the proponent identified below, with a copy to eng@smpte.org.

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:

Scott Smyers
DTS, Inc.
130 Knowles Drive, Suite B
Los Gatos, CA 95032

Email: scott.smyers@dts.com

Approved
August 23, 2016

Table of Contents

Page

Conventions

All sections are normative, unless otherwise indicated. Pseudo-code and property names use font style `courier new`.

The expressions MAY, NEED NOT, SHALL, SHALL NOT, SHOULD, and SHOULD NOT indicate normative behavior as specified in ISO/EIC Directives, Part 2. Edition 6.0, 2011-04.

VERBAL FORM	SEMANTICS
MAY	It is allowed
NEED NOT	It is not required that
SHALL	Is required that
SHALL NOT	Is required to be not
SHOULD	It is recommended that
SHOULD NOT	It is not recommended that

Introduction (Informative)

The Bitstream consists of an ordered sequence of Packets. As illustrated in Figure 1, each Packet consists of a payload preceded by a Label field identifying the nature of the payload and a length field indicating the size of the payload. Packet are byte-aligned, but information within their payloads not necessarily so. Implementations can skip over Packets without knowledge of their payload, enabling straightforward extensibility.

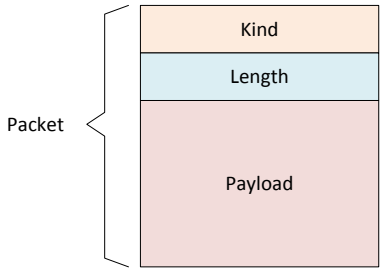

Figure 1 – Packet

Packets are logically grouped into hierarchical structures. Specifically, as depicted in Figure 2, a Bitstream consists of a sequence of Frame structures, each containing the metadata and audio samples required to completely reproduce a Program for a specified interval within its timeline. In particular, each Frame structure contains a complete copy of the Program header information, thereby allowing playback to start on any Frame structure boundary without requiring access to prior or future Frames.

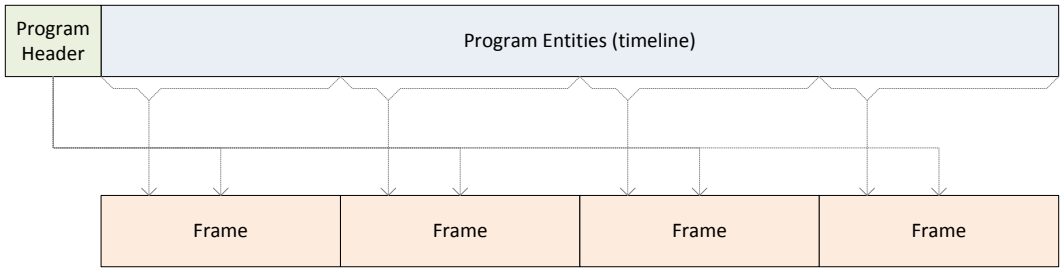

Figure 2 – Mapping Program to Bitstream Frame Structures

As illustrated in Figure 3, Frame structures are further segmented, with Program header information followed by Fragment structures, each containing Entity metadata for a specified time interval within the timeline of the Frame. The audio samples associated with the Frame are contained in a sequence of Asset Frame Packets.

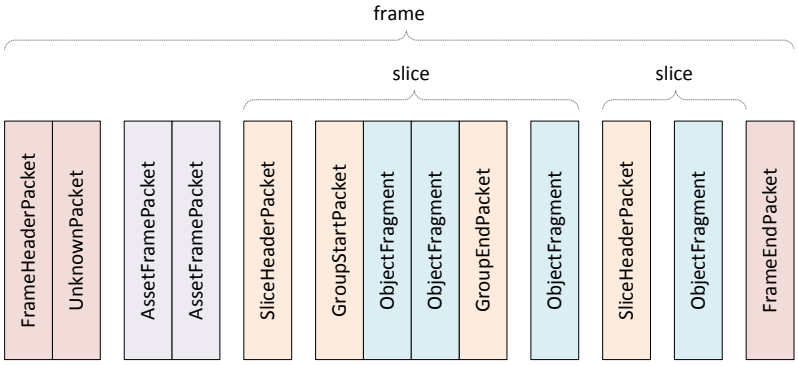

Figure 3 – Frame Structure

The Program object model makes extensive use of URI as unique identifiers. To reduce overhead, this specification defines mappings between common URI values and shorter Label values.

1 Scope

This specification maps the MDA Program specified in the MDA Program Specification, including audio samples, to a single binary structure – the Bitstream. The Bitstream is partitioned into Frames such that reproduction can start at any Frame boundary, without knowledge of earlier or future Frames.

2 Normative References

MDA Program Specification 1.03

Internet Engineering Task Force (IETF) (January 2005). RFC 3986 – Uniform Resource Identifier (URI): Generic Syntax

ISO/EIC Directives, Part 2. Edition 6.0, 2011-04

Internet Engineering Task Force (IETF) (January 2003). RFC 3629 – UTF-8, A Transformation Format of ISO 10646

3 Structures

The Bitstream syntax is expressed using the operation of a hypothetical parser using the structure specification language described in Section 10.

For extensibility, the Bitstream syntax allows the presence of unknown Packets – captured by fields of type `UnknownPacket`. Implementations MAY ignore these unknown Packets.

3.1 Bitstream

A Bitstream consists of an ordered sequence of Frames, as specified in Table 1.

Each Frame is an item of the `fFrame[]` collection. The number of Frames in an MDA Bitstream for a given MDA Program SHALL be less than or equal to 2^{64}.

Table 1 – Bitstream Structure

```
aligned struct Bitstream {

 var long unsigned int i = 0;

 while(!eof) {

  peek PacketHeader fUnknownHeader;

  // Frames

  if (fUnknownHeader.fKind == FrameHeaderPacket.fKind) {
   Frame fFrame[i++];
  }

  // Unknown packets
```