SMPTE ENGINEERING REPORT

35PM Study – Automated Processing with IMF OPL for TSP 2121 AMWA AS-11 Applications

SMPTE ER 1006:2021

The home of media professionals, technologists, and engineers.

SMPTE

SMPTE Engineering Report

35PM Study – Automated Processing with IMF OPL for TSP 2121 AMWA AS-11 Applications

Table of Contents

Page

Approved **March 24, 2021**

1 Scope

The report identifies certain IMF post-processing use cases and provides example solutions using IMF Output Profile List (OPL) technology. The example solutions make use of existing OPL standard macros and also propose new macros as required to provide a complete solution in OPL vernacular.

2 Introduction

The Interoperable Master Format (IMF) framework SMPTE ST 2067 was created to standardize packaging, interchange, and processing of media content using componentized media encodings. Interoperability between implementations is based on core constraints, which are further specialized for a number of applications. A Composition Playlist (CPL) is an IMF structure that defines the content of a media playback timeline, but it only defines the time of playback of each content element. To then process the essence from the IMF composition timeline into a derivative format, e.g., to re-size a moving-image sequence or assign audio channel destinations, the Output Profile List (OPL) specifies the parameters of a set of processing nodes, or Macros, that transform the CPL timeline essence.

The broadcast community has shown keen interest in the use of automation to prepare content derivatives from an IMF-encoded library. OPL is designed to support this general use case. The Digital Production Partnership (DPP) is an initiative to help producers and broadcasters maximize the potential benefits of digital television production. SMPTE Study Group (SG) participants have brought forward for study a set of use cases developed by the DPP that represent current practice and ongoing initiatives for IMF interchange and IMF-based automated content provisioning. This report identifies real use cases for the broadcast community and it is hoped that the development of the suggested interoperable OPL macros will encourage adoption among broadcast automation vendors.

2.1 OPL Workflow

An OPL document describes the signal processing to be performed on the image, sound and other essence and metadata that are carried in an IMF composition to create a derivative that might or might not be IMF.

An OPL document is associated with a single instance of an IMF Composition Playlist (CPL). The CPL provides the source essence to be transformed by the OPL processor. This tight coupling allows the OPL contents to rely on the specific coding properties of the composition, which in turn simplifies the expression and analysis of a given OPL. The OPL processor decodes the information in the OPL document, CPL document and respective IMF Track Files and produces the described derivative composition as output.

An unlimited number of OPL documents may be associated with a given CPL. This flexibility allows a single CPL to be used to create as many derivative variants as need be described using the available OPL vocabulary. Private namespaces in OPL are supported, allowing implementers to extend the standard processing model to accommodate unique processing elements while taking advantage of the overall OPL framework. Managing the complexity of these potential combinations is beyond the scope of this study.

2.2 General Usage

The following general purpose operations were discussed throughout the study effort:

1. OPL documents can be used by a content producer to establish normative processing requirements that can then be communicated to multiple partners in an interoperable way, e.g., a peculiar Slope-Offset-Power (SOP) function, a Look-up Table (LUT), a particular audio channel routing and labeling scheme;

2. Although not addressed in this report, an OPL image compositing macro could lead to a further reduction of duplicate inventory for certain localization use cases;

3. Validation tools can compare process results with independently calculated check frames to verify precision;

4. Service providers can communicate processing "actuals" in the form of an OPL document back to the customer, allowing the customer to, e.g., verify that the processing was performed as expected or to refer to a previous setup as the basis for new work.

2.3 OPL Fundamentals

An Output Profile List (OPL) is an XML document that describes a signal processing chain for audio/visual signals. The ultimate purpose of a given OPL instance is to describe the signal paths between the selected virtual tracks of an IMF Composition Playlist (CPL) and the input channels of some kind of output device or file. Every sample of the timeline defined by the CPL will be transformed by the macros in the OPL into some kind of output: baseband samples to a monitoring device, a multiplexed codestream suitable for broadcast, etc. An OPL cannot alter the timeline of a composition (that is the purpose of the CPL).

A "macro" is a virtual signal processing node in an OPL. Macros are linked together into a directed graph by way of identifiers that allow assignment of macro inputs from CPL virtual tracks and the outputs of other macros. As hinted above, a complete OPL instance terminates in an output macro which defines a process for writing a file or otherwise transmitting the result of the process. The use cases defined in this report are limited to creating files (as will be described in more detail below).

The simplest possible OPL contains a single macro which selects virtual tracks and produces the output signal. Typically, however, additional macro nodes are required to reformat the image and route the audio channels. Much of the discussion of the use cases below consists of the signal processing steps required to transform the signal in a given IMF composition into a format suitable for use by the output macro.

For simplicity, the signal processing domain of all image macros operating with Reference Pixels is 4:4:4[:4]. Where chroma subsampling is required, it is expected that the pixel encoder and/or decoder will perform the appropriate operation. Similarly, the signal processing domain of all audio macros is 24-bit linear Pulse-code Modulation (PCM) at the sample rate identified by the virtual track essence.

2.4 OPL Granularity

OPL Macro items are intended to provide the low-level, elemental operators needed to translate a wide (and widening) variety of signal processing standards into precise runtime instructions for transcoders and other decoder applications. While such fine granularity may lead to very complex OPL instances, it will be shown that this complexity is beneficial, and relatively straightforward to address where it might otherwise be difficult to manage.

The first area of concern is user interface. It is supposed that complicated macros will be difficult to create and interpret by hand. This will be true in most cases, but it is expected that software will mediate all applications and will therefore be available to isolate the user from unnecessary detail, provide local constraints on the available operations, support process visualization, etc. Rarely, for example, does an end user edit a CPL by hand; software tools are required to support high-level use.

The second area of concern is in the image processor itself, where much effort has been expended by manufacturers to provide high performance given available resources, often taking advantage of limitations in scope (i.e., some steps can be skipped if you know how) or customer expectations (no one wants to pay for precision they will not use.) Manufacturers are rightfully wary of processing rules that are wasteful of resources. In this area it is important to be aware that an IMF OPL processor need not execute the macros in an OPL in the precise order in which they are specified.

An IMF OPL processor can implement any portion of a given OPL using any optimized process that combines or otherwise interprets individual Macro items, so long as the result is congruent with the normative interpretation of that OPL. Said another way, an implementation can analyze an OPL to compose a suitable process from its available operators, for example, substituting a LUT for a matrix operator. Such a system is compliant insofar as it produces the same or similar result as that of the literal interpretation of that same OPL, within tolerances appropriate for that application.

What is gained at this level of detail is the flexibility to define a very wide variety of processes without requiring explicit support from the underlying application. In addition, these process definitions will be portable, and the result of an operation defined by one will be comparable to a reference standard, and will thus support process debugging by providing precise configuration detail. Having fewer and more simple macros will also simplify the task of defining and implementing interoperable applications that will create and interpret OPL documents.

3 Reference Documents

SMPTE RP 177:1993, Derivation of Basic Television Color Equations

SMPTE ST 2067-2:2020, Interoperable Master Format – Core Constraints

SMPTE ST 2067-3:2020, Interoperable Master Format – Composition Playlist

SMPTE ST 2067-21:2020, Interoperable Master Format – Application #2E

Amendment 1:2021 to SMPTE ST 2067-21:2020

SMPTE ST 2067-100:2014, Interoperable Master Format – Output Profile List

SMPTE ST 2067-101:2018, Interoperable Master Format – Output Profile List – Common Image Definition and Macros

SMPTE ST 2067-102:2017, Interoperable Master Format – Common Image Pixel Color Schemes

SMPTE ST 2067-103:2021, Interoperable Master Format – Output Profile List – Common Audio Definition and Macros

SMPTE TSP-2121-1:2018, Interoperable Master Format – Application DPP (ProRes) (https://www.smpte.org/technical-specifications/tsp2121-app-dpp)

Note: The decision of SMPTE to replace TSP-2121-1:2018 by the Registered Disclosure Document (RDD) 59-1 was approved during the development of this report. Reference from SMPTE TSP-2121-1:2018 to SMPTE RDD 59-1 will be accessible after the publication of SMPTE RDD 59-1.

SMPTE TSP-2121-4:2019, Interoperable Master Format – Application Constraint DPP (JPEG2000) (https://f.hubspotusercontent00.net/hubfs/5253154/tsp2121-4-2019.pdf)

Note: The decision of SMPTE to replace TSP-2121-4:2019 by the Registered Disclosure Document (RDD) 59-2 was approved during the development of this report. Reference from SMPTE TSP-2121-4:2019 to SMPTE RDD 59-2 will be accessible after the publication of SMPTE RDD 59-2.

AMWA AS-11 UK DPP HD v1.1
http://amwa-tv.github.io/AS-11_UK_DPP_HD/AMWA_AS_11_UK_DPP_HD.html

AMWA AS-11 X1
http://amwa-tv.github.io/AS-11_X1/AMWA_AS_11_X1.html

EBU Technical Recommendation R48-2005, Allocation of audio tracks on digital television recorders
https://tech.ebu.ch/docs/r/r048.pdf

EBU Recommendation R 123 (July 2009), EBU Audio Track Allocation for File Exchange https://tech.ebu.ch/docs/r/r123.pdf

ISO/IEC 14496 – ISO BMFF Byte Stream Format

ITU-R BT.2390-8 (02/2020), High dynamic range television for production and international programme exchange

iTunes Package TV Specification 5.3.3
https://help.apple.com/itc/tvspec/

iTunes Package Film Specification 5.3.4
https://help.apple.com/itc/filmspec/

BBC Perceptual Quantiser (PQ) to Hybrid Log-Gamma (HLG) Transcoding

BBC Subtitle Guidelines
https://bbc.github.io/subtitle-guidelines/

4 Definitions

4.1 Runtime Property

A property of the OPL processing environment that is determined in an implementation-specific way, at the time of OPL processing. This property could be a value fixed by the application, obtained from a configuration, requested from the user, etc. Note that runtime properties are not present in an OPL instance document; they are used in this report to identify essential properties that are explicitly left to the implementation.

4.2 Parametric Property

A property of the OPL processing environment that is embedded directly in the output macro, i.e., a parameter to a Macro instance.

4.3 External Property

A property of the OPL processing environment that is taken from the input CPL or from an externally referenced metadata source.

4.4 CPL Timeline

The complete composition timeline as described in the Composition Playlist (CPL), which consists of a set of virtual tracks selected from the CPL.

4.5 TSP 2121

Includes SMPTE TSP 2121-1 (ProRes™) and SMPTE TSP 2121-4 (Jpeg2000) unless qualified further.

5 Use Case Template

Each use case will be presented using the idiom provided in this section. For each use case, there will be a brief characterization of the work to be performed and the types of compositions that are appropriate as input. A detailed description will then follow, including all nodes in the OPL signal path and all properties required for execution in an interoperable OPL processing environment.

5.1 Description

A short description of the transformation to be performed, i.e., the output to be produced.

5.2 Input Range

The allowable variations in the input CPL. This description notes the IMF Application and any applicable constraints.

5.3 Output

A prose description of the output artifact(s) to be produced.

5.4 Process

5.4.1 Image Processing

A schematic representation of an image processing signal path. If multiple paths are possible, they will be iterated.

Each component in a chain of OPL image macros is represented by a text symbol, with optional parametric properties indicated in succeeding parentheses, for example ColorTransform3x3(<npm-name>). The components are stated left-to-right, with the input on the left and output on the right. Between each pair of adjacent macros a symbol is present to indicate the numerical domain in which samples are transferred between the respective macros. The following symbols are currently defined for image essence:

Table 1. Symbol definitions for image essence

Symbol	Description
~>	Non-linear Reference Pixels
=>	Linear Reference Pixels

As an example, consider Figure 1:

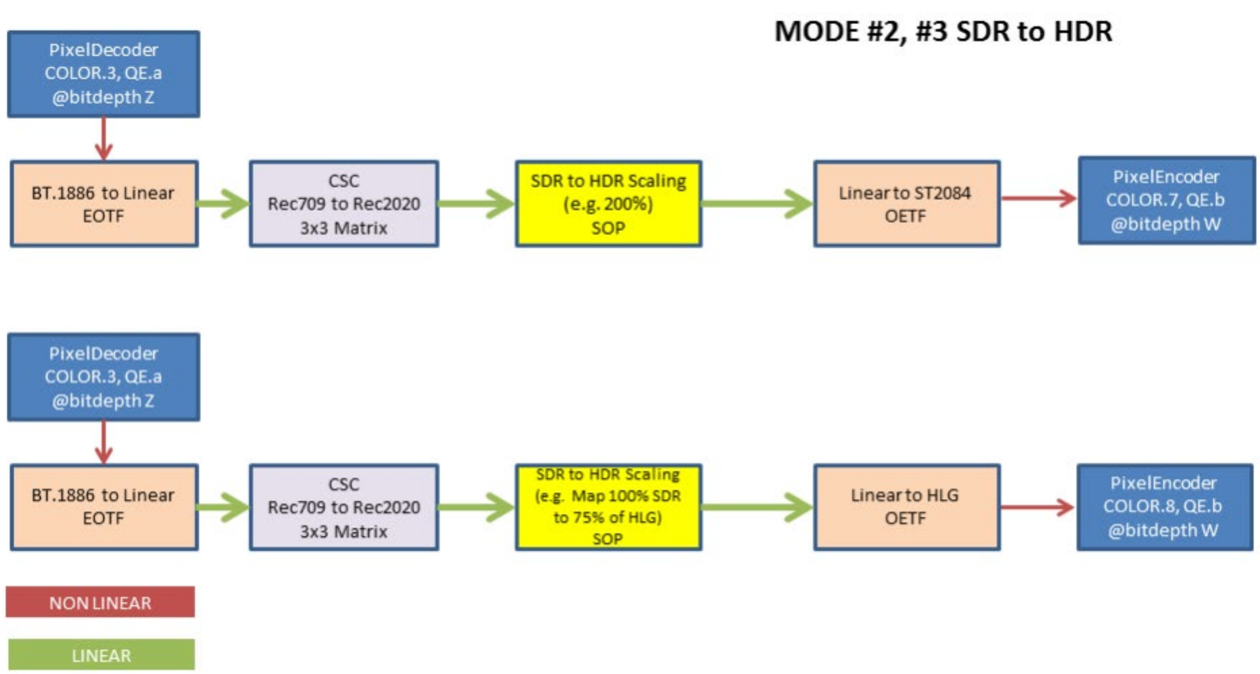

Figure 1. Schematic syntax for mode #2, #3 SDR to HDR

The two image processing modes in Figure 1 are described in 6.2.4.1. In these processes, Standard Dynamic Range (SDR) images are converted to High Dynamic Range (HDR) images. In the context of this document, SDR images have peak luminance of 100 candela per meter square and luminance contrast range of less than 2000:1. In addition, HDR images can have peak luminance up to 10,000 candela per meter square and a luminance contrast range far exceeds that of SDR.

The notation of COLOR.n in this document is specified by IMF Application #2E per SMPTE ST 2067-21 and its amendment. Each integer value for “n” in the range [3-8] represents a combination of color primary system and transfer characteristics.

Using the schematic syntax presented above, the first macro chain would be represented as follows:

```
Decoder(COLOR.3) ~> EotfBt1886
       => ColorTransform3x3(Npm709To2020)
       => SlopeOffsetPower(2,0,1)
       => Oetf2084 ~> Encoder(COLOR.7)
```

and the second macro chain would be represented as follows:

```
Decoder(COLOR.3) ~> EotfBt1886
       => ColorTransform3x3(Npm709To2020)
       => SlopeOffsetPower(0.75,0,1)
       => OetfHlg ~> Encoder(COLOR.8)
```

While not as easily recognized at first when compared to the image, the schematic syntax has the following advantages:

- the text is easily managed by source control;
- the text can be pasted into program source as documentation;
- hyperlinks may be embedded in the text.

5.4.2 Audio Processing

The output macros identified in this report are proposed as having individual audio input channels, allowing the routing of a channel of a CPL virtual track directly to the macro input without the need for any intervening audio processing macro.

If summing or static gain control is required, then one or more instances of the AudioRoutingMacro can be instantiated. In this case a schematic representation of the audio processing signal path must be provided. If multiple paths are possible, they can be iterated using a table.

Each component in a chain of OPL audio macros is represented by a text symbol, with optional parametric properties indicated in succeeding parentheses, for example `AudioRouting`. The components are stated left-to-right, with the input on the left and output on the right. Between each pair of adjacent macros a symbol is present to indicate the numerical domain in which samples are transferred between the respective macros. The following symbols are currently defined for audio essence:

Table 2. Symbol definition for audio essence

Symbol	Description
->	Single-channel PCM Audio

5.4.3 Additional Processing

Additional sections may be created to describe selection and/or transformation of data types other than image and audio.

5.5 Output Macro

This is a subsection grouping of the properties of the output macro identified for this application.

5.5.1 Output Macro Name

A symbol that identifies the output macro definition.

5.5.2 Image Source

The source of images to the output macro. This value is either a CPL virtual track handle or the name of the output macro node that terminates the image processing macro chain. A macro can be designed to accept more than one image source.

5.5.3 Audio Source

The source of audio samples to the output macro. This value is either a CPL virtual track-channel handle or the name of the output macro node that terminates the audio processing macro chain. A macro will typically be designed to accept more than one audio source.

5.5.4 Timed Text Source

The source of timed text data to the output macro. This value is a CPL virtual track handle that references Timed Text Markup Language (TTML) essence.

5.5.5 User Metadata Source

The source of the DPP metadata object required for AS-11 assembly. At this time, the `DM_AS_11_Segmentation_Framework` structure is embedded in the CPL in the `ExtensionProperties` element. The output macro processor is expected to retrieve the data item from the CPL and supply it to the output macro.

5.5.6 Runtime Properties

Values provided by the runtime environment, i.e., not present in the OPL instance document. The most common example is an output filename used by an output macro.

5.5.7 Parametric Properties

Properties of the output macro that control its behavior by e.g., selecting operational profile, defining metadata items, etc.

6 DPP Use Cases for OPL

The following is the list of use cases presented by the stakeholders in the DPP IMF work group:

1. Use Case 1: Re-wrap Essence to .mov
2. Use Cases 2,3: Transcode TSP 2121 to AMWA AS-11 X1
3. Use Cases 4,5: Transcode TSP 2121 to AMWA AS-11 UK DPP HD

Each use case adapts a particular SMPTE TSP 2121-1 IMF composition to a particular output variant. In the case of .mov output the result is a single QuickTime™ file containing ProRes™ images and PCM audio. In the other cases an AMWA AS-11 MXF file is created, containing images and audio as required by the AMWA AS-11 X1 and UK DPP HD specifications. It is also possible in any case to create a text file containing TTML text extracted from a TTML virtual track in the IMF composition.

6.1 Use Case 1: Re-wrap Essence to .mov

6.1.1 Description

Produce a QuickTime™ .mov file from a ProRes™ SMPTE TSP 2121 composition. The OPL will select the composition's virtual image track directly, implying that the implementation can optimize the process by simply copying the ProRes codestreams directly from the source track files into the output file. PCM audio channels in the IMF composition can be selected into numbered audio channels in the output file.

Note that this use case does not support transformation from 4:4:4 to 4:2:2 or vice-versa (i.e., the output file will have the same sampling structure as the input composition because it is really the same codestream.) To perform either operation would require processing the baseband image which is beyond the scope of this use case.

6.1.2 Input Range

SMPTE TSP 2121 25p/50p 4:2:2/4:4:4, ProRes™ only, COLOR.3, COLOR.7, and COLOR.8, in the following formats.

Table 3.

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10, 12, 16	QE.1, QE.2
COLOR.7	10, 12, 16	QE.1, QE.2
COLOR.8	10, 12	QE.1, QE.2

Output color spaces are determined by the input.

6.1.3 Output

Produce a QuickTime™ .mov file that contains the image and audio content of the entire SMPTE TSP 2121 composition timeline. Samples in the output file are copied directly from the respective IMF track files.

6.1.4 Process

6.1.4.1 Image Processing

None

6.1.4.2 Audio Processing

Optional summing or gain control via `AudioRoutingMixing` macro.

6.1.5 Output Macro

6.1.5.1 Output Macro Name

`IsoBmffOutputMacroType`

6.1.5.2 Image Source

CPL Virtual Track

6.1.5.3 Audio Source

CPL Virtual Track or from Audio Routing Macro(s)

Several inquiries were made by the study group to collect delivery specifications for QuickTime™ .mov files but little was received. At this time there is no known requirement for additional metadata related to the presentation of PCM audio channels in .mov files.

6.1.5.4 Runtime Properties

- `OutputPath`

6.1.5.5 Parametric Properties

None

6.1.6 Example

The following illustrates the use of the proposed `IsoBmffOutputMacroType` macro. The example demonstrates use of the `AudioRoutingMixing` macro to create a composite audio channel.

Table 4. Use of the proposed IsoBmffOutputMacroType

```
<OutputProfileList
    xmlns="http://www.smpte-ra.org/schemas/2067-100/2014"
    xmlns:arm="http://www.smpte-ra.org/schemas/2067-103/2014"
    xmlns:opl-ng="urn:cinecert.com:new-opl-stuff"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Id>urn:uuid:ec8ef34a-d7a2-47b2-b524-d10738626beb</Id>
  <Annotation>Re-wrap TSP-2121 to Quicktime(TM)</Annotation>
  <IssueDate>2020-06-15T15:05:00+00:00</IssueDate>
  <Issuer>jh</Issuer>
  <Creator>jh</Creator>
  <CompositionPlaylistId>urn:uuid:0b55db4d-c918-447a-8c3d-
                   26a2c3b76c7c</CompositionPlaylistId>
  <AliasList/>
  <MacroList>

    <!-- Sum the channels of one of the tracks into a single output channel "out". --
                   >
    <Macro xsi:type="arm:AudioRoutingMixingMacroType">
      <Name>node-lr-sum</Name>
      <arm:OutputEntityList>
        <arm:OutputAudioChannel>
          <arm:Handle>out</arm:Handle>
          <arm:InputEntityList>
            <arm:InputEntity>
              <arm:Handle>cpl/virtual-tracks/fef03f4d-1c71-4f66-8f0c-
                   29485d8f85b8?MCATagSymbol=chL</arm:Handle>
              <arm:Gain>-3.0</arm:Gain>
            </arm:InputEntity>
            <arm:InputEntity>
              <arm:Handle>cpl/virtual-tracks/fef03f4d-1c71-4f66-8f0c-
                   29485d8f85b8?MCATagSymbol=chR</arm:Handle>
              <arm:Gain>-3.0</arm:Gain>
            </arm:InputEntity>
          </arm:InputEntityList>
        </arm:OutputAudioChannel>
      </arm:OutputEntityList>
    </Macro>

    <Macro xsi:type="opl-ng:IsoBmffOutputMacroType">
      <Name>qt-output</Name>
      <Annotation>Use Case 0: Re-wrap TSP-2121 to Quicktime</Annotation>
```

```
      <opl-ng:OutputPath>https://objectstore.example.com/example-as-11-X1-
                   output.mxf</opl-ng:OutputPath>

      <opl-ng:InputReferenceImageSequence>
        <!-- The image codestreams may be copied directly from the
             input track files, the processor need not process the
             image.  -->
        <opl-ng:Handle>cpl/virtual-tracks/b3766b5a-77cd-4e3b-818b-50110d8caaf9</opl-
                   ng:Handle>
      </opl-ng:InputReferenceImageSequence>

      <opl-ng:AudioInputEntityList>
        <!-- 5.1 main sound track, default labels carry through -->
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex>0</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chL</opl-ng:Handle>
        </opl-ng:InputEntity>
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex>1</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chR</opl-ng:Handle>
        </opl-ng:InputEntity>
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex>2</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chC</opl-ng:Handle>
        </opl-ng:InputEntity>
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex>3</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chLFE</opl-ng:Handle>
        </opl-ng:InputEntity>
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex >4</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chLS</opl-ng:Handle>
        </opl-ng:InputEntity>
        <opl-ng:InputEntity>
          <opl-ng:ChannelIndex>5</opl-ng:ChannelIndex>
          <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chRS</opl-ng:Handle>
        </opl-ng:InputEntity>
      </opl-ng:AudioInputEntityList>
    </Macro>
  </MacroList>
</OutputProfileList>
```

6.2 Use Cases 2,3: Transcode TSP 2121 to AMWA AS-11 X1

6.2.1 Description

Programs commissioned as SMPTE TSP 2121-1 are required to be distributed as AMWA AS-11 X1 YUV 4:2:2.

6.2.2 Input Range

SMPTE TSP 2121 25p/50p 4:2:2, COLOR.3, COLOR.7, and COLOR.8, in the following formats.

Table 5. Input color spaces for TSP 2121 25p/50p 4:2:2

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10, 12, 16	QE.1, QE.2
COLOR.7	10, 12, 16	QE.1, QE.2
COLOR.8	10, 12	QE.1, QE.2

SMPTE TSP 2121 25p/50p 4:4:4, COLOR.3, COLOR.7, and COLOR.8, in the following formats.

Table 6. Input color spaces for TSP 2121 25p/50p 4:4:4

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10, 12, 16	QE.1, QE.2
COLOR.7	10, 12, 16	QE.1, QE.2
COLOR.8	10, 12	QE.1, QE.2

6.2.3 Output

`AMWA_AS_11_OutputMacro`

Table 7. Output color spaces

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10	QE.1
COLOR.7	10	QE.1
COLOR.8	10	QE.1

6.2.4 Process

6.2.4.1 Image Processing

For each of the nine possible input/output pairs a macro chain is suggested to accomplish required transformation.

Table 8. Macro chains for Use Case #2

Mode	Description
1	`Decoder(COLOR.3) ~> Encoder(COLOR.3)`
2	`Decoder(COLOR.3) ~> EotfBt1886 => ColorTransform3x3(Npm709To2020) => SlopeOffsetPower(2,0,1) => Oetf2084 ~> Encoder(COLOR.7)`
3	`Decoder(COLOR.3) ~> EotfBt1886 => ColorTransform3x3(Npm709To2020) => SlopeOffsetPower(2,0,1) => OetfHlg ~> Encoder(COLOR.8)`
4	`Decoder(COLOR.7) ~> Eotf2084 => SlopeOffsetPower(0.5,0,1) => ColorTransform3x3(Npm2020To709) => OetfBt1886 ~> Encoder(COLOR.3)`
4a	`Decoder(COLOR.7) ~> Eotf2084 => LUT1d(LocalMagic) ~> Encoder(COLOR.3)`
5	`Decoder(COLOR.7) ~> Encoder(COLOR.7)`
6	`Decoder(COLOR.7) ~> Eotf2084 => OetfHlg ~> Encoder(COLOR.8)`
7	`Decoder(COLOR.8) ~> EotfHlg => LUT3d(BbcInvHlg) ~> Encoder(COLOR.3)`
8	`Decoder(COLOR.8) ~> EotfHlg => Oetf2084 ~> Encoder(COLOR.7)`

9	`Decoder(COLOR.8) ~> Encoder(COLOR.8)`

Notes:

- The left-most element in each chain receives pixels decoded from the track file(s);
- The SlopeOffsetPower macro is present to provided scaling from SDR to HDR (modes 2 and 3) and from HDR to SDR (modes 5, 6, 8, and 9.)

6.2.4.2 Audio Processing

None

6.2.5 Output Macro

6.2.5.1 Output Macro Name

`AMWA_AS_11_OutputMacro`

6.2.5.2 Image Source

Pixel Encoding Macro

6.2.5.3 Audio Source

CPL Virtual Track or from Audio Routing Macro(s)

6.2.5.4 User Metadata Source

CPL Extension Properties

6.2.5.5 Runtime Properties

- `OutputPath`
- `PackageUID`

6.2.5.6 Parametric Properties

Table 9. List of parametric properties for use case #2, #3

Property	Value
`AS-11-Profile`	AMWA_AS_11_X1
`AudioChannelFormat`	One of the values defined in AMWA-AS-11 Output Macro
`DM_AS_11_Segmentation_Framework`	A regXML expression containing the required segmentation values

6.3 Use Cases 4,5: Transcode TSP 2121 to AMWA AS-11 UK DPP HD

6.3.1 Description

Programs commissioned as SMPTE TSP 2121-1 required to be distributed as AMWA AS-11 UK DPP HD.

6.3.2 Input Range

SMPTE TSP 2121 25p/50p 4:2:2, COLOR.3, COLOR.7, and COLOR.8, in the following formats.

Table 10. Input color spaces for TSP 2121 25p/50p 4:2:2

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10, 12, 16	QE.1, QE.2
COLOR.7	10, 12, 16	QE.1, QE.2
COLOR.8	10, 12	QE.1, QE.2

Table 11. Input color spaces for TSP 2121 25p/50p 4:4:4

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10, 12, 16	QE.1, QE.2
COLOR.7	10, 12, 16	QE.1, QE.2
COLOR.8	10, 12	QE.1, QE.2

6.3.3 Output

`AMWA_AS_11_OutputMacro`

Table 12. Output color spaces

Colorimetry	Pixel Bit Depth	Quantization
COLOR.3	8, 10	QE.1

6.3.4 Process

6.3.4.1 Image Processing

The following transformations are required.

Table 13. Macro chains for Use Case #4

Mode	Description
1	`Decoder(COLOR.3) ~> Encoder(COLOR.3)`
2	`Decoder(COLOR.7) ~> Eotf2084 => SlopeOffsetPower(0.5,0,1) => ColorTransform3x3(Npm2020To709) => OetfBt1886 ~> Encoder(COLOR.3)`
3	`Decoder(COLOR.8) ~> EotfHlg => LUT3d(BbcInvHlg) ~> Encoder(COLOR.3)`

6.3.4.2 Audio Processing

None

6.3.5 Output Macro

6.3.5.1 Output Macro Name

`AMWA_AS_11_OutputMacro`

6.3.5.2 Image Source

Pixel Encoding Macro

6.3.5.3 Audio Source

From Audio Routing Macros(s)

6.3.5.4 User Metadata Source

CPL Extension Properties

6.3.5.5 Runtime Properties

- `OutputPath`
- `PackageUID`

6.3.5.6 Parametric Properties

Table 14. List of parametric properties for use case #4, #5

Property	Value
`AS-11-Profile`	AMWA_AS_11_UK_DPP_HD
`AudioChannelFormat`	One of the values defined in AMWA-AS-11 Output Macro
`DM_AS_11_Segmentation_Framework`	A regXML expression containing the required segmentation values

7 Existing OPL Macros

7.1 PixelDecoderType

An instance of the `PixelDecoderType` macro, defined in SMPTE ST 2067-101, produces a sequence of images composed of reference pixels, where each output pixel is derived from a corresponding coded pixel in the input essence. The macro input is taken from the uncompressed form of images stored in an IMF composition. In the use cases studied in this document the images will be as defined in TSP-2121, i.e., decoded from ProRes™.

The `PixelDecoderType` is responsible for up-sampling the input as required to produce a 4:4:4[:4] reference image.

The transformation function, from the uncompressed essence samples into the Reference Pixel domain, is determined by essence descriptor values associated with the MXF structure that contains the source images. Specifically, the following properties of the MXF Generic Picture Essence Descriptor are relevant:

- PixelBitDepth;
- ComponentMinRef and ComponentMaxRef signal restricted or full range scales;
- the colorimetry label (e.g., COLOR.3, COLOR.8) may be determined from TransferCharacteristic, CodingEquations, and ColorPrimaries.

Incidentally, another MXF Generic Picture Essence Descriptor property, PictureEssenceCoding, is how the decoder knows which codec to use when converting from stored images to uncompressed images.

7.2 PixelEncoderType

An instance of the `PixelEncoderType` macro, defined in SMPTE ST 2067-101, produces a sequence of images composed of coded pixels, where each output pixel is derived from a corresponding reference pixel received via the input handle. The macro encodes reference pixels into the quantization and sample format signaled by its ColorEncoding property.

7.3 AudioRoutingMixingMacroType

The `AudioRoutingMixingMacroType`, defined in SMPTE ST 2067-103, provides facilities for routing and combining PCM audio channels.

8 Proposed OPL Macros

8.1 ColorTransform3x3MacroType

An instance of the proposed `ColorTransform3x3MacroType` macro produces a sequence of images composed of reference pixels, where each output pixel is derived from a corresponding reference pixel received via the input handle, multiplied by the given constant matrix.

Table 15. Example Color Transform 3x3 Macro

```
<Macro xsi:type="opl-ng:ColorTransform3x3Type">
  <Name>node-3</Name>
  <opl-ng:InputReferenceImageSequence>
    <opl-ng:Handle>macros/node-2/outputs/images</opl-ng:Handle>
  </opl-ng:InputReferenceImageSequence>
  <opl-ng:OutputReferenceImageSequence>
    <opl-ng:NormalizedPrimaryMatrix>
      0.4851 0.2501 0.0227
      0.3488 0.6977 0.1162
      0.1302 0.0521 0.6860
    </opl-ng:NormalizedPrimaryMatrix>
  </opl-ng:OutputReferenceImageSequence>
</Macro>
```

8.2 LookupTable3dMacroType

An instance of the proposed LookupTable3dMacroType macro produces a sequence of images composed of reference pixels, where each output pixel is derived from a corresponding reference pixel received via the input handle, transformed by performing the lookup procedure described next.

To support the use of integer LUTs in a reference pixel environment, it is proposed that the LookupTable3dMacroType converts the reference pixel input value into a specified integer domain prior to applying LUT to the pixel. The resulting value is then converted to a reference pixel, again using a transformation function identified in the macro instance.

Consider the following example:

Table 16. Example Lookup Table 3d Macro

```
<Macro xsi:type="opl-ng:LookupTable3dMacroType">
  <Name>img-eotf</Name>
  <opl-ng:InputReferenceImageSequence>
    <opl-ng:ColorEncoding>http://www.smpte-ra.org/schemas/2067-102/2014#REC709-
              RGB-10</opl-ng:ColorEncoding>
    <opl-ng:Handle>macros/img-decoder/outputs/images</opl-ng:Handle>
  </opl-ng:InputReferenceImageSequence>
  <opl-ng:OutputReferenceImageSequence>
    <opl-ng:ColorEncoding>http://www.smpte-ra.org/schemas/2067-102/2014#REC709-
              RGB-10</opl-ng:ColorEncoding>
    <opl-ng:InterpolationMethod>trilinear<opl-ng:InterpolationMethod>
    <opl-ng:LookupTable3d>
      <!-- use LUT to implement EOTF -->
      4096 4096 4096
      <!-- ... -->
      60160 60160 60160
```

```
      </opl-ng:LookupTable3d>
    </opl-ng:OutputReferenceImageSequence>
  </Macro>
```

In this case the use case is to apply a LUT pre-computed for use with 10-bit RGB pixels. By configuring the macro to interpret the input reference pixels as 10-bit RGB, and then converting the lookup result back to reference pixels, the integer LUT is supported.

The LUT interpolation method was also considered, and it was recommended that trilinear and tetrahedral methods be supported.

8.3 LookupTable1dMacroType

This macro type is similar to the 3D LUT macro discussed above, but only 1D using linear interpolation.

8.4 SlopeOffsetPowerMacroType

An instance of the proposed `SlopeOffsetPowerMacroType` macro produces a sequence of images composed of reference pixels, where each output pixel is derived from a corresponding reference pixel received via the input handle, transformed by performing the procedure described next.

Note that for simplicity and (hopefully) familiarity, this proposal relies on the American Society of Cinematographers (ASC) Color Decision List (CDL) capability of the same name.

The macro is configured with three sets of constant values, where each set consists of Slope, Offset, and Power, one set for each component in the input pixel.

For each input pixel, the macro instance will produce an output pixel by performing the following operations on each component (except alpha), using the respective constant set:

- multiply the component value by the “Slope” constant;
- add the “Offset” constant to the component value;
- exponentiate the component value (base) with the “Power” constant (exponent).

Table 17. Example Slope Offset Power Macro

```
  <!-- SOP -->
  <Macro xsi:type="opl-ng:SlopeOffsetPowerMacroType">
    <Name>img-sop</Name>
    <opl-ng:InputReferenceImageSequence>
      <opl-ng:Handle>macros/img-npm/outputs/images</opl-ng:Handle>
    </opl-ng:InputReferenceImageSequence>
    <opl-ng:OutputReferenceImageSequence>
      <asc-cdl:ColorDecisionList>
        <asc-cdl:ColorDecision>
          <asc-cdl:ColorCorrection id="user-value">
            <asc-cdl:SOPNode>
              <asc-cdl:Slope>1.0 1.0 1.0</asc-cdl:Slope>
              <asc-cdl:Offset>0.0 0.0 0.0</asc-cdl:Offset>
              <asc-cdl:Power>1.0 1.0 1.0</asc-cdl:Power>
            </asc-cdl:SOPNode>
          </asc-cdl:ColorCorrection>
        </asc-cdl:ColorDecision>
      </asc-cdl:ColorDecisionList>
    </opl-ng:OutputReferenceImageSequence>
  </Macro>
```

8.5 EotfBt1886Macro

Encodes linear reference pixels as Recommendation ITU-R BT.1886 values using the transfer function defined in Recommendation ITU-R BT.1886.

Table 18. Example EOTF BT.1886 Macro

```
<Macro xsi:type="opl-ng:SMPTETransferCharacteristicType">
  <Name>linear to Rec709</Name>
  <opl-ng:InputReferenceImageSequence>
    <opl-ng:Handle>macros/node-2/outputs/images</opl-ng:Handle>
  </opl-ng:InputReferenceImageSequence>
  <opl-ng:OutputReferenceImageSequence>
    <opl-ng:SMPTELabelDictionaryID>
      urn:smpte:ul:060e2b34.04010101.04010101.01020000
    </opl-ng:SMPTELabelDictionaryID>
    <opl-ng:Direction>OETF</opl-ng:Direction>
  </opl-ng:OutputReferenceImageSequence>
</Macro>
```

8.6 OetfBt1886Macro

Decodes Recommendation ITU-R BT.1886 values as linear reference pixels using the transfer function defined in Recommendation ITU-R BT.1886.

Table 19. Example OETF BT.1886 Macro

```
<Macro xsi:type="opl-ng:SMPTETransferCharacteristicType">
      <Name>Rec709 to linear</Name>
      <opl-ng:InputReferenceImageSequence>
        <opl-ng:Handle>macros/node-2/outputs/images</opl-ng:Handle>
      </opl-ng:InputReferenceImageSequence>
      <opl-ng:OutputReferenceImageSequence>
        <opl-ng:SMPTELabelDictionaryID>
          urn:smpte:ul:060e2b34.0401010d.04010101.010a0000
        </opl-ng:SMPTELabelDictionaryID>
        <opl-ng:Direction>EOTF</opl-ng:Direction>
      </opl-ng:OutputReferenceImageSequence>
    </Macro>
```

8.7 Eotf2084Macro

At the time of writing there is not enough information available to specify a dedicated macro for this transfer function. Implementations can use a LUT or a proprietary macro to define the required processing.

8.8 Oetf2084Macro

At the time of writing there is not enough information available to specify a dedicated macro for this transfer function. Implementations can use a LUT or a proprietary macro to define the required processing.

8.9 EotfHlgMacro

At the time of writing there is not enough information available to specify a dedicated macro for this transfer function. Implementations can use a LUT or a proprietary macro to define the required processing.

8.10 OetfHlgMacro

At the time of writing there is not enough information available to specify a dedicated macro for this transfer function. Implementations can use a LUT or a proprietary macro to define the required processing.

8.11 IsoBmffOutputMacroType

Write the composition timeline as a QuickTime™-compatible ISO/IEC 14496 - ISO BMFF Byte Stream Format (ISOBMFF) "movie" file using ProRes™ and PCM codecs for image and sound, respectively.

8.11.1 BMFF Formulation

The output of the macro is a file written to a location provided by the caller. The contents of the file will be data items as defined in ISO 14496. The arrangement and contents of these data items must satisfy the requirements of portable QuickTime™ files as currently defined by Apple Inc.

8.11.2 Image Input Source

A pixel encoder macro or virtual track. Note that the implementation can optimize the image path in the case where the image track codec is ProRes™, by simply forwarding the unwrapped ProRes™ codestreams to the output file.

8.11.3 Audio Input Source

The macro writes PCM audio samples as it receives them from the inputs listed in the `AudioInputEntityList` element. Each item in the list is an `InputEntity`, which contains an OPL handle identifier and one or more optional properties that provide additional metadata for the audio source. The OPL handle can be a virtual track selector or a handle created by another macro such as `AudioRoutingMixingMacroType`.

8.11.4 Runtime Properties

- `OutputPath`

8.11.5 Example

See Clause 6.1 Use case 1.

The example in Table 4 illustrates a configuration having three soundfields: one 5.1 main, and one each stereo and mono alternative. The mono alternative is constructed by summing the left and right channels of the source virtual track using an instance of the `AudioRoutingMixingMacroType` macro.

8.12 TtmlFileOutputMacroType

Write TTML text to a file. Text is copied verbatim from the source track into the output file.

The virtual track must be homogeneous, a single Internet Media Subtitles and Captions (IMSC) profile must be in use throughout the track. The use of IMSC profile in IMF is specified in SMPTE ST 2067-2.

In the case where a virtual track is comprised of edit units from more than one track file, the macro writes a single document instance containing the complete set of text instances from the combined input stream.

Multiple TTML documents (i.e., coming from different track files on the same virtual track) can be merged into a single TTML document using the <div> element. The TTML and IMSC Test Suites contain examples using multiple <div> elements that are used to partition the TTML document into several sections.

The use-case that motivated the design of this macro is strictly based on text-only virtual tracks, i.e., no image profiles are used in the track files. Moreover, the general practice is to rely on generic font families rather than custom font families to avoid issues with readability and unexpected font family replacement mechanisms (fallback). Because of these constraints, the current design of the macro will not output font resources stored in the MXF track files.

For more guidance on TTML/IMSC subtitles authoring in the context of this macro, please refer to the document BBC Subtitle Guidelines.

Note on image profiles and virtual tracks with mixed text/image profiles:

At the time of writing the study group has recognized that one might want to use the TtmlFileOutputMacroType outside the scope of the use-case. In situations where the TTML/IMSC documents wrapped in the MXF track files make use of additional resources such as Portable Network Graphics (PNG) files and fonts, a good practice would be to extract those resources in the same folder where the macro generated the main TTML document. However, this has severe limitations as several track files can use different resources resolving to the same PNG or font resource file name. To solve this problem, a more elaborated mechanism involving full parsing and resource name replacement/substitution is necessary.

In addition, it is important to notice that some IMF applications do not put constraints on the homogeneity of the timed text virtual tracks. While IMSC does not support a "mixed" profile yet, one could create a TTML document using another profile that allows image and text to be mixed.

8.12.1 Data Input Source

Select a TTML input from virtual track.

8.12.2 Runtime Properties

- `OutputPath`

8.12.3 Example

Table 20. Example TTML File Output Macro

```
<!-- All edit units from a timed text virtual track written to a single TTML
                 document instance. -->
<Macro xsi:type="opl-ng:TtmlFileOutputMacroType">
  <Name>closed-cap</Name>
  <opl-ng:DocumentPath>
    https://objectstore.example.com/example-text-output.ttml.xml
  </opl-ng:DocumentPath>
  <opl-ng:InputEntityList>
    <opl-ng:InputEntity>
      <opl-ng:Handle>
        cpl/virtual-tracks/162b4c21-6e76-4436-b9e7-50cb7a3c204f
    </opl-ng:Handle>
    </opl-ng:InputEntity>
  <opl-ng:InputEntityList>
</Macro>
```

8.13 AMWA_AS_11_OutputMacro

Write the composition timeline as a constrained AMWA AS-11 MXF file. The macro supports two constraints identifiers.

Table 21.

Symbol	Reference
AMWA_AS_11_UK_DPP_HD	MXF Program Contribution - UK DPP HD
AMWA_AS_11_X1	MXF Program Contribution - DPP UHD

The macro will determine the application constraints to be signaled in the output file by observing the contents of an AS-11-Profile property.

8.13.1 DPP Metadata

A metadata object in the CPL (defined in DPP003 Carriage of User Metadata in SMPTE TSP 2121 could be selected by this output macro via, e.g., xpath.) Additional metadata is also taken from CPL elements such as ContentTitle.

8.13.2 Segmentation Metadata

AMWA AS-11 files might contain segmentation metadata which indicate temporal ranges of the composition's timeline and associated metadata items. This information appears in the AS-11 output file as a Program Segmentation Track, which is implemented as an MXF Descriptive Metadata Track. This construction is defined in both UK DPP and X1.

The application that creates the OPL instance can copy the `DM_AS_11_Segmentation_Framework` element from the CPL into the OPL instance, or it can construct a custom `DM_AS_11_Segmentation_Framework` for the specific instance.

8.13.3 Image Input Source

The image will be taken from the output of the Encoder macro that terminates the image processing macro chain.

8.13.4 Audio Input Source

The macro has numbered channel inputs which can select any channel in a virtual track or the output of an `AudioRoutingMixingMacroType` macro. The macro will signal the channel format to the output file, based on the AudioChannelFormat parametric property. The channel formats supported by the macro are based on the following channel format identifiers:

Table 22. Channel format identifiers

Symbol	Description
EBU-R-48:2a	Stereo
EBU-R-123:4b	Stereo with M&E
EBU-R-123:4c	Stereo with Audio Description
EBU-R-123:16c	Stereo, 5.1 & M&E (with & without Audio Description)
EBU-R-123:16d	5.1 Two Languages
EBU-R-123:16e	Three Languages

The symbols in Table 22 are proposed by the SG as a starting point for the audio input sources. The document numbers “EBU-R-48” and “EBU-R-123” are the European Broadcast Union (EBU) Recommendations that specify the listed audio formats. The decimal with alphabet designates the number of channels in an audio layout defined by the EBU Recommendations. For example, “2a” is the two channels layout “a” in EBU-R-48, “4b” is the four channels layout “b” in EBU-R-123, and “16c” is the 16 channels layout “c” in EBU-R-123.

Note: The use of “:” in the proposed symbols in Table 22 could be problematic in some encodings; it is use in a standard symbolic syntax must be studied carefully by the standards developers.

For DPP HD, tokens are based on the table of references in AMWA_AS_11_UK_DPP_HD.

For AS-11-X1, tokens are based on the table of references in AMWA_AS_11_X1.

The output will assume the SMPTE ST 377-4 MCA label of the source essence unless that value is overridden in the `InputEntity`. In the case of a macro source such as `AudioRoutingMixingMacroType`, Audio channel routing is symbolic and positional: the macro will assign `InputEntity` items to output channels in the order in which they are present in the `InputEntityList` element.

It is not required to provide a source for each numbered input. The macro inserts silent channels as needed to make a compliant output stream.

8.13.5 Data Input Source

Select TTML input from a virtual track.

8.13.6 Runtime Properties

- `OutputPath` (an URI value)

8.13.7 Example

Table 23. Example AMWA AS-11 Output Macro

```
<Macro xsi:type="opl-ng:AMWA_AS_11_OutputMacroType">
  <Name>as-11-output</Name>
  <Annotation>Use Case 2: Conform YUV to AMWA AS-11 X1 HLG</Annotation>

  <opl-ng:OutputPath>
    https://objectstore.example.com/example-as-11-X1-output.mxf
  </opl-ng:OutputPath>
  <opl-ng:PackageUID>
urn:smpte:umid:060a2b34.01010107.01010f20.13000000.f124ad03.8d5c4971.8dab1c2e.0f9875f
  </opl-ng:PackageUID>
  <opl-ng:AS_11_Profile>AMWA_AS_11_X1</opl-ng:AS_11_Profile>
  <opl-ng:AudioChannelFormat>AudioLayoutMode-0</opl-ng:AudioChannelFormat>

  <as11:DM_AS_11_Segmentation_Framework>
    <r0:AS_11_Part_Total>1</r0:AS_11_Part_Total>
    <r0:AS_11_Part_Number>1</r0:AS_11_Part_Number>
    <r1:LinkedDescriptiveFrameworkPluginID>
      urn:uuid:d9a4afd3-b904-464e-ba84-b9481f4984f9
    </r1:LinkedDescriptiveFrameworkPluginID>
  </as11:DM_AS_11_Segmentation_Framework>

  <opl-ng:InputReferenceImageSequence>
    <opl-ng:Handle>macros/img-encoder/outputs/images</opl-ng:Handle>
  </opl-ng:InputReferenceImageSequence>

  <opl-ng:AudioInputEntityList>
    <!-- 5.1 main sound track, default labels carry through -->
    <opl-ng:InputEntity>
      <opl-ng:ChannelIndex>0</opl-ng:ChannelIndex>
      <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                230730592766?MCATagSymbol=chL</opl-ng:Handle>
    </opl-ng:InputEntity>
    <opl-ng:InputEntity>
      <opl-ng:ChannelIndex>1</opl-ng:ChannelIndex>
      <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                230730592766?MCATagSymbol=chR</opl-ng:Handle>
    </opl-ng:InputEntity>
    <opl-ng:InputEntity>
      <opl-ng:ChannelIndex>2</opl-ng:ChannelIndex>
      <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                230730592766?MCATagSymbol=chC</opl-ng:Handle>
    </opl-ng:InputEntity>
    <opl-ng:InputEntity>
      <opl-ng:ChannelIndex>3</opl-ng:ChannelIndex>
```

```
        <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chLFE</opl-ng:Handle>
      </opl-ng:InputEntity>
      <opl-ng:InputEntity>
        <opl-ng:ChannelIndex >4</opl-ng:ChannelIndex>
        <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chLS</opl-ng:Handle>
      </opl-ng:InputEntity>
      <opl-ng:InputEntity>
        <opl-ng:ChannelIndex>5</opl-ng:ChannelIndex>
        <opl-ng:Handle>cpl/virtual-tracks/77ab4eae-cb47-520e-8cbf-
                   230730592766?MCATagSymbol=chRS</opl-ng:Handle>
      </opl-ng:InputEntity>
    </opl-ng:AudioInputEntityList>
  </Macro>
```

The output will assume the SMPTE ST 377-4 MCA label of the source essence unless that value is overridden in the `InputEntity`. In the case of a macro source such as `AudioRoutingMixingMacroType` the label must be specified.

9 Named Lookup Tables

A Lookup Table (LUT) can be used in the place of mathematical computation for a fixed set of input values. Common examples are color space conversion and color correction functions. In the OPL framework, any LUT can be documented with a specific name. The following examples for a color transcoding and a customized "magic" transform illustrate the use of Named Lookup Tables.

9.1 BbcInvHlg

Inverse HLG transform per BBC Perceptual Quantizer (PQ) to Hybrid Log-Gamma (HLG) Transcoding.

9.2 LocalMagic

A made-up name that illustrates the use of local knowledge to reduce some set of operations to a LUT.

10 Named Normalized Primary Matrices

The Normalized Primary Matrices (NPM) as defined in SMPTE RP 177 are essential to color space conversion operations. In the OPL framework, any 3x3 NPM can be documented with a specific name. The following examples for converting between Recommendation ITU-R BT.709 and Recommendation ITU-R BT.2020 color primary systems illustrate the use of Named Normalized Primary Matrices.

10.1 Npm709To2020

Transform linearized Recommendation ITU-R BT.709 reference pixels into linearized Recommendation ITU-R BT.2020 reference pixels.

$$NPM = \begin{bmatrix} 1.6604 & -0.5876 & -0.0728 \\ -0.1245 & 1.1328 & -0.0083 \\ -0.0181 & -0.1005 & 1.1187 \end{bmatrix}$$

10.2 Npm2020To709

Transform linearized Recommendation ITU-R BT.2020 reference pixels into linearized Recommendation ITU-R BT.709 reference pixels.

$$NPM = \begin{bmatrix} 0.6274 & 0.3292 & 0.0433 \\ 0.0690 & 0.9195 & 0.0113 \\ 0.0163 & 0.0880 & 0.8955 \end{bmatrix}$$