EBU – Recommendation R 121

Material Exchange Format
Basic User Metadata Implementation

EBU Recommendation

Source: P/TV-FILE

Geneva
February 2007

Page intentionally left blank. This document is paginated for recto-verso printing

Scope and applicability of this recommendation

This recommendation contains a list of basic provisions for MXF implementations.

These provisions constitute necessary conditions for reliable, system-wide application of user Metadata that is carried in the header Metadata of MXF files or that is linked to the Contents of MXF files via header Metadata identifiers.

All of these provisions can be implemented by manufactures regardless of the version, type or nature of the user Metadata plug-in.

This recommendation does not specify how to apply the plug-in and linkage mechanisms, or which descriptive Metadata schemes to use for certain applications.

The EBU encourages users and implementers to follow this recommendation for file interchange, for implementation in their production systems and for implementing their products.

Conformance Notation

This document contains both **normative** text and **informative** text.

All text is normative except for that in the Introduction, any section explicitly labelled as 'Informative' or individual paragraphs that start with '*Note*:'.

Normative text describes indispensable or mandatory elements. It contains the conformance keywords '*shall*', '*should*' or '*may*', defined as follows:

'*Shall*' and '*shall not*':	Indicate requirements to be followed strictly and from which no deviation is permitted in order to conform to the document.
'*Should*' and '*should not*':	Indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others. OR indicate that a certain course of action is preferred but not necessarily required. OR indicate that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
'*May*' and '*need not*'	Indicate a course of action permissible within the limits of the document.

Informative text is potentially helpful to the user, but it is not indispensable and it can be removed, changed or added editorially without affecting the normative text. Informative text does not contain any conformance keywords.

A conformant implementation is one that includes all mandatory provisions ('shall') and, if implemented, all recommended provisions ('should') as described. A conformant implementation need not implement optional provisions ('may') and need not implement them as described.

Page intentionally left blank. This document is paginated for recto-verso printing

Contents

Page intentionally left blank. This document is paginated for recto-verso printing

Material Exchange Format

Basic User Metadata Implementation

EBU Committee	*First Issued*	*Revised*	*Re-issued*
PMC	2006		

Keywords: MXF, Material Exchange Format, User Metadata, Descriptive Metadata

Introduction

An appealing feature of MXF is its flexible support of synchronised user Metadata that is carried in the header Metadata of MXF files or that is linked to the contents of MXF files via header Metadata identifiers.

These recommendations concerning the application of user Metadata in MXF have been deliberated by EBU Project Group P/TV-FILE after discussions with MXF implementers and vendors.

Carriage of user Metadata in MXF files

User Metadata can be carried in MXF by the following means:

1) In MXF header Metadata
   a) using the SMPTE 377M Descriptive Metadata plug-in mechanism;
   b) using proprietary extensions of MXF structural Metadata sets;
2) In generic container system item elements;
3) In specific areas of essence streams (e.g. MPEG-2 Video ES user bits)

Each of the three carriage mechanisms supports a different set of features. Further details are given in Annex B.

Linkage of external user Metadata to MXF files

MXF header Metadata uses globally unique identifiers such as the SMPTE 330M *Unique Material Identifier* (UMID), the ISO-IEC 11578 *Universal Unique Identifier* (UUID) or identifiers that are unique within a well-defined local context (such as Track ID values within a Package).

The values of these identifiers can be used:

1) to link between essence and Metadata stored within MXF files,
2) to link between the material stored in MXF files and external Metadata that could be stored in databases or Metadata files,
3) to link between Metadata stored in MXF files and external Metadata that could be stored in databases or Metadata files.

Recommendations

The EBU recommends the following implementation parameters in order to support the application of user Metadata within MXF environments:

Recommendation for store and forward operations

1) Store and forward applications shall preserve all user Metadata.

2) Store and forward applications shall preserve

   a) all globally unique identifiers (GUID),

   b) all locally unique identifiers that are used for the MXF source reference mechanisms, and

   c) all timecode information.

3) Store and forward applications that employ lossy transcoding of essence or that change the essence (such as partial copy) shall update the UMID according to SMPTE RP205.

Recommendation for processing MXF files

4) Applications that generate new MXF files from one or more MXF input files (e.g. partial copy or editing applications) shall use the source reference mechanism to support tracking of material and associated descriptive Metadata (e.g. user Metadata).

   a) Material tracking shall be implemented such that the essence track source clips of the top-level source packages of the new file reference the package and track in the original file.

   b) The application shall permit the configuration of the essence track source reference mechanism such that the source references point to either:

      i) the material packages of the original files

      Note: This can be interpreted as 'recording' a virtual 'play-out' of the material package of the original file. It is needed to preserve linkage to Metadata that has been associated to the output timeline of the original file either via the header Metadata plug-in mechanism or linked via header Metadata ID values in external databases. When converting the material package of the original file into a lower level source package of the new file, this method also allows to preserve linkage to Metadata that has been associated to the stored material of the original file either via the header Metadata plug-in mechanism or linked via header Metadata ID values in external databases.

      ii) the top level source packages of the original files

      Note: This can be interpreted as applying an EDL to extract portions of the original essence containers. It may not preserve linkage to Metadata that has been associated to the output timeline of the original file, but preserves linkage to Metadata that has been associated to the stored material of the original file either via the header Metadata plug-in mechanism or linked via header Metadata ID values in external databases.

   c) Lower-level source packages created using the source reference mechanism should be kept internal to the file.

      Note: This is needed to transparently support descriptive Metadata in certain production flows, especially where user Metadata is generated during the production process and the temporary MXF files and their Metadata are not registered with a Content management system.

   d) Users shall be able to turn off the functionality to keep lower-level source packages internal to the new MXF file.

      Note: This is needed in order to filter (suppress) user Metadata in the new file.

Recommendation for user Metadata applications based on SMPTE 380M

5. Applications that define Metadata sets with SMPTE 380M-syntax shall identify themselves via a unique ID value in the framework thesaurus name property of the production, clip and scene framework sets.

   Note: In this context, applications are primarily user applications such as the Eurvision News Exchange or the newsroom system of a specific broadcaster and not a specific MXF application provided by a specific vendor.

   a) The unique ID value shall be a SMPTE UL that is registered in the SMPTE Labels Registry (SMPTE 400M/RP224)

   b) The unique ID value shall use the 'urn:oid' string encoding specified in RFC 3061.

   Note: The motivation for these requirements is to enable receivers of such Metadata to discriminate between different applications that use Metadata sets with SMPTE 380M-syntax.

6. In addition to the Universal Labels that identify the SMPTE 380M frameworks, the DM Schemes property of the Preface shall also contain application specific Universal Labels.

Documentation recommendation for vendors

7. As part of their product documentation, vendors should provide detailed documentation about the parameters and functionalities of the timecode implementation [see Annex C].

Bibliography

SMPTE 377M-2004 Television - Material Exchange Format (MXF) - File Format Specification

SMPTE 330M-2004 Television - Unique Material Identifier (UMID)

SMPTE 379M-2004 Television - Material Exchange Format (MXF) - MXF Generic Container

SMPTE 385M-2004 Television - Material Exchange Format (MXF) -Mapping SDTI-CP Essence and Metadata into the MXF Generic Container

SMPTE 394M-2005 Television - Material Exchange Format (MXF) - System Scheme 1 for the MXF Generic Container

ISO/IEC 11578 Information technology - Open Systems Interconnection - Remote Procedure Call (RPC)

SMPTE RP 205 2000 Application of Unique Material Identifiers in Production and Broadcast Environments

SMPTE 380M-2004 Television - Material Exchange Format (MXF) – Descriptive Metadata Scheme-1

SMPTE 400M-2004 Television - SMPTE Labels Structure

SMPTE RP 224 Television - SMPTE Labels Registry

(IETF) RFC 3061 A URN Namespace of Object Identifiers

EBU BPN 071 EBU MXF Implementation tests, May 2006 (EBU Members only)

Page intentionally left blank. This document is paginated for recto-verso printing

Annex A: User requirements for the support of user Metadata in MXF implementations

The EBU has identified the following user requirements for the implementation of user Metadata in MXF files. These requirements abstract from specific user Metadata schemes. They focus on the baseline functionality that is needed in order to transport and interchange user Metadata in MXF-based systems.

1) Store and forward applications need to be transparent to user Metadata carried in MXF files.

2) Store and forward applications need to be transparent to user Metadata stored in external databases.

3) Applications which generate new MXF files from one or more input MXF files (i.e. partial restore, editing) need to be implemented such that it is possible to find Metadata which was associated to the original material.

4) Basic implementation of user Metadata needs to be independent of the specific user Metadata scheme or schemes that are used by an organisation.

5) When exporting files, it needs to be possible to eliminate all user Metadata and source material links.

6) SMPTE 380M defines a general framework to represent and encode user Metadata. Users apply these frameworks by specifying the precise use and semantic for their application. In order to reliably use and merge user Metadata in environments where multiple applications generate and use SMPTE 380M-defined sets, the DMS-1 instances need to identify the application specification.

Page intentionally left blank. This document is paginated for recto-verso printing

Annex B: Features of the three Metadata carriage mechanisms

B1. Carriage through MXF Header Metadata

The carriage through MXF Header Metadata can be implemented in two ways: either via the SMPTE 377M Descriptive Metadata plug-in mechanism, or via proprietary extensions of MXF Structural Metadata sets.

The first method implies that a set of special Metadata sets (descriptive Metadata frameworks) is defined or chosen among the available standards (e.g. DMS-1) and instantiated in the Header Metadata of MXF files. Frameworks can be directly linked to the Structural Metadata sets of MXF via the standard plug-in mechanism. Designers of user applications can choose whether to associate the frameworks on static Metadata tracks (i.e. Metadata are valid globally) or on timeline or event tracks (i.e. specifying to which temporal portion of the material the Metadata refer to), depending on the specific application purpose. Every MXF decoder should be able to identify the presence of the linked frameworks and may be able to extract them from the Header Metadata and present them to the driving application without keys translation and data decoding. Sub-sets referred inside the frameworks may not be recognised and extracted. If standard descriptive Metadata frameworks are used, compliant decoders should be able to identify the complete structure of Metadata (frameworks and linked sets), to decode the keys against the semantic labels and data from their data-encoding pattern.

The second method implies that proper extensions of the structural Metadata sets defined for MXF are defined and implemented (DM Segment and/or DM Source Clip), by e.g. adding new propertys or new structures (i.e. properties constituting strong references and hierarchies of Metadata sets referenced by those properties) to them. Not all MXF decoders may be able to identify extended structural sets, given that the KLV keys for these extended sets shall be different from the ones defined in the standard for the basic (not extended) sets. If different applications extend a set by the same property (i.e. SMPTE Metadata dictionary elements identified by the same SMPTE UL), but with different semantics (i.e. meaning of the property in the context of the set) successful interchange of information will not be possible.

B2. Carriage through generic container system item elements

This technique allows applications to store descriptive Metadata in the Generic Container System Item Elements [see SMPTE 385 and SMPTE 394]. With this mechanism, potentially each Content Package (e.g. each video frame) can be associated to a specific instance of Metadata. Metadata can be encoded in Global Sets, Local Sets or Packs. Each System Item Element is represented by a distinct set or pack. With the use of multiple elements in the System Item, different data sets can be associated to the same Content package at the same time. Specifications must be written to define the use of these sets inside the System Item. Basic MXF decoders may be able to access System Item elements inside the Essence Container Streams and to select those referring to the different kinds of essence tracks via the Track Number information.

B3. Carriage in specific areas of essence streams

This technique allows the carriage of Metadata items encoded in some specific areas of the essence streams, e.g. MPEG-2 Video Elementary Stream User bits. These Metadata cannot be accessed by any basic MXF decoder, and specific extensions to basic elementary stream decoders (e.g. MPEG video decoders) may be needed to recover Metadata from the encoded stream.

Page intentionally left blank. This document is paginated for recto-verso printing

Annex C: Documentation Template for vendors

1. Store and forward Applications

a. Does the application preserve all user Metadata? If not, which user Metadata is preserved and which user Metadata is lost? (specify)	
b. Does the application preserver all globally unique identifier? If not, which identifiers are preserved and which identifiers are modified? (specify)	
c. Does the application preserve all locally unique identifiers that are used for the MXF source reference mechanism? If not, which identifiers are preserved and which identifiers are modified? (specify)	
d. Does the application preserve all timecode information? If not, which Timecode information is preserved and which Timecode information is lost? (specify)	
e. If the application employs lossy transcoding of Essence, or changes the Essence, does it update the Package ID values (i.e. UMIDs) accordingly?	

2. Processing MXF Files

a. Does the application support material tracking via the source reference mechanism?	
b. Which of the source reference mechanism configuration options does the application support [see 4)b)]? (specify)	
c. Does the application support (i.e. create and maintain) file-internal lower level source packages? (specify)	
d. It is possible to turn off (i.e. suppress or filter) lower level source packages? (specify)	

3. Metadata Applications based in SMPTE 380M

a. Is the (user-) application identified by a SMPTERP 224 registered UL value in the framework thesaurus property of the production, clip and scene framework sets? If not, how is the application identified? (specify)	
b. How is the application identifier encoded? (specify)	
c. Is the application identifier value also listed in the DM Schemes property of the Preface set?	